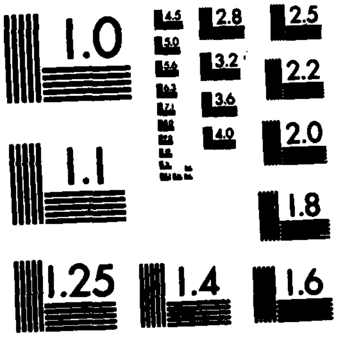
AD-A124 212   EDITSPEC: SYSTEM MANUAL VOLUME VIII PROGRAMMERS MANUAL   1/1
(U) CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY)
CHAMPAIGN IL E S NEELY OCT 79 DOD/DF-83/002J

UNCLASSIFIED   F/G 9/2   NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

EDITSPEC SYSTEM MANUAL

VOLUME 8: PROGRAMMERS MANUAL

by

E. S. Neely

October 1979

Department of the Army
CONSTRUCTION ENGINEERING RESEARCH LABORATORY
P.O. Box 4005
Champaign, IL 61820

77

UNITED STATES DEPARTMENT OF COMMERCE
National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Date 4 Feb 1983 NTIS Control # 035042

TO: Defense Technical Information
Center - DTIC
Cameron Station
Alexandria, Virginia 22314

FROM: NTIS, Input Branch
5285 Port Royal Road
Springfield, Virginia 22161

Report # DOD-DF-83-0423 ADA #

Title: EDITSPEC - Sig Manual V 8

Subject report is ☒ Standard Process □ STG report. STG - Special Technology Group. □ Computer Product □ Follow up date

□ The report will be accessioned by DDC. The form noting the ADA numb is returned.

□ The report has been assigned the ADA number noted above and is retu to NTIS for processing.

□ DDC will not process the report. It is returned to NTIS

Mag Tape Price
PC & MF Price
Source DOD A
Stock Quantity
Source Share

Comments:

Dottie Adams
Signature (for billing)

Copy when completed to
Finance Branch

DOD Report Action Request
(Replaces NTIS-164 5-72)

Please return to NTIS - Information Services Branch, Room 301, Yorktown Building.
Attention: Dottie Adams

REPORT DOCUMENTATION PAGE

1. REPORT NO.
DOD/DF-83/002j

2.

3. Recipient's Accession No.
AD-A124212

4. Title and Subtitle
EDITSPEC: System Manual, Volume VIII: Programmers Manual

5. Report Date
October 1979

6.

7. Author(s)
Edgar S. Neely, Jr.

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

Department of the Army
Construction Engineering Research Laboratory
P.O. Box 4005
Champaign, IL 61820

10. Project/Task/Work Unit No.
4A762731AT41/T1/009

11. Contract(C) or Grant(G) No.
(C)
(G)

12. Sponsoring Organization Name and Address

(same)

13. Type of Report & Period Covered

14.

15. Supplementary Notes

For magnetic tapes, see

16. Abstract (Limit: 200 words)

The EDITSPEC System is an automated system designed to produce construction specifications from Corps of Engineers Guide Specifications. The System uses one central computer and a communications network to provide remote terminal access by Corps offices, nationwide to a central data base.

This volume provides programmers with the procedures to master the EDITSPEC system and to add new features to the system. EDITSPEC's internal character representation and command processing are presented first. The standard application commons and subroutines are described next. The required access methods to tables are then given. The last chapter provides the suggested method for adding new functional commands to the system.

17. Document Analysis a. Descriptors

Construction Specifications
Guide Specifications
Military Construction

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement:

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page)
UNCLASSIFIED

21. No. of Pages

22. Price

ABSTRACT

This volume provides programmers with the procedures to master the EDITSPEC system and to add new features to the EDITSPEC system. EDITSPEC's internal character representation and command processing is presented first. The standard application commons and subroutines are described next. The required access methods to tables are then given. The last chapter provides the suggested method for adding new functional commands to the system.

CONTENTS

Page

APPENDICES

1. MASTERING THE EDITSPEC SYSTEM

This chapter attempts to guide the programmer who knows nothing about the EDITSPEC system into a position of understanding the logic of the system design and being ready to begin program design and implementation.

The best way to obtain a working knowledge of the system is to understand the functions that the system is currently performing. The programmer should read the first few chapters of the users manual in detail and then review the purpose of each command in the remaining chapters.

The basic programmer's tools are FORTRAN, the data handler, and the table handler. The programmer should already have several years of experience in coding production programs in FORTRAN. FORTRAN manuals should be obtained for reference. All codes should follow the standards of ANSI FORTRAN.

A solid working knowledge of the data handler and table handler should be obtained next by reading Volumes IV and then III of the EDITSPEC System Manual. Both systems are basic tools needed to perform coding for EDITSPEC.

The programmer should then start to learn the system design concepts by reading the "Construction Specification Preparation Within the EDITSPEC System" report. Volume 1, Systems Overview, should be read next followed by Volume 2, System Design Concepts.

Detailed programming instructions are given in the remaining sections of this manual.

2. EDITSPEC CHARACTER REPRESENTATION

The actual representation of a character in machine code and the sequential order of the machine character representation are entirely machine dependent. The EDITSPEC system uses one consistent internal code to represent all allowable characters. This internal code is shown in Table 2.1. where

The first 26 internal characters identify all special characters recognized by the system. Special characters are identified as punctuation or nonpunctuation characters as shown in the table. A word within the text of a document may have punctuation characters before and/or after the characters in the word. For example, the word "house" may appear as "(house)" in the text. The punctuation characters must be overlooked when trying to locate the word "house."

The next 26 characters with internal codes 27 through 52 represent the alphabetic character set. EDITSPEC internal representation represents lower-case alphabetic characters as the characters themselves. Upper-case alphabetic characters are represented by two internal characters. The first is an internal code of 63 to indicate that the next character is to be printed as an upper-case character. The second character is the lower-case representation of the upper-case character.

The two-character notation was adopted to minimize computer searching time when trying to locate all occurrences of the word "house." The text may contain house in several forms:

1. house

2. House

3. HOUSE

4. hoUSE

5. HOuse

Text searches are normally performed in all lower case with the internal code 63 ignored during comparison.

The internal codes 53 through 62 represent the characters "0" through "9."

When the user enters a character that is not contained in the internal code, the characters will be translated to a question mark (?).

3. EDITSPEC COMMAND PROCESSING

The FORTRAN subroutine CMMND controls the processing of all commands. CMMND calls another subroutine GTCMD to get the next command. GTCMD calls subroutine RDLIN to read the command from an external device. RDLIN calls one of four subroutines to perform the actual reading of the command.

The /PARSC/ common is used to store all related information about the command. The machine character representation of the command is stored in array MCLIN in "A1" format for machine-independent writing. Array STRNG contains the command in EDITSPEC internal codes in a packed format.

The commands are decoded by the subroutine DECOD. Array PMPTR contains the total number of parameters in the command in PMPTR (1, 3). The actual information about each parameter is stored in the rows of PMPTR. The parameter types and PMPTR information is given in Table 2.2, Decode Variables.

The DECODE subroutine calls subroutine MATCH to identify the type of command that has been entered. MATCH compares the command name string entered with the allowable command name acronyms and sets variable ISUB to point to the correct subroutines for CMMND to transfer control to for actual processing of the command itself.

When the command processing routine returns control to CMMND, CMMND will start the process all over again.

Preceding page blank

SUBROUTINE DECOD

PROGRAM: SUBROUTINE DECOD

FUNCTION: INTERPRET A COMMAND AND CALL THE PROPER SUBROUTINE

AUTHOR: PETER KARP

MODIFICATIONS:

LANGUAGE: FORTRAN

CALLING SEQUENCE: CALL DECOD

TASKS OR MODULES:

VARIABLES:

```
        CHARY - ARRAY WHICH CONTAINS UNPACKED COMMAND STRING
        ICHAR - INDEX INTO CHARY (POINTS TO CURRENT CHARACTER)
        ISUB  - VALUE USED IN COMPUTED GO TO DETERMINE WHICH
                COMMAND SUBROUTINE TO CALL
        KOUNT - USED TO CALCULATE EACH PARAMETER SIZE
        NCHAR - MAXIMUM NO. OF CHARACTERS IN THIS COMMAND STRING
        MAXST - MAXIMUM ALLOWABLE STRING LENGTH
        NCHST - COUNTER FOR CURRENT LENGTH OF CURRENT STRING
        PMPTR - A 2-DIMENSIONAL ARRAY WHICH CONTAINS THE DESCRIPTION
                OF ALL PARAMETER FIELDS FOUND FOR THE PRESENT COMMAND
           PMPTR (1,1) = NOT USED
           PMPTR (1,2) = NOT USED
           PMPTR (1,3) = THE NO. OF PARAMETER FIELDS FOUND FOR THIS
                         COMMAND
           PMPTR (N,1) = THE CHARACTER LOCATION WITHIN THE COMMAND
                         STRING OF THE FIRST CHARACTER OF THIS
                         PARAMETER
           PMPTR (N,2) = THE LENGTH OF THIS PARAMETER (NO. OF CHAR)
           PMPTR (N,3) = PARAMETER TYPE
        PTYPE - TYPE OF PARAMETER

                                  1 - NUMERIC
                                  2 - ALPHANUMERIC STRING
                                      (NOT ENCLOSED WITHIN DELIMITERS)
                                  3 - ALPHANUMERIC STRING
                                      (ENCLOSED WITH DELIMITERS)
                                  4 - SEMICOLON
                                  5 - HYPHEN
                                  6 - COMMA
                                  7 - DOUBLE ASTERISK
                                  8 - PLUS
                                  9 - COLON

        PRMNO - THE CURRENT PARAMETER NO.
        PRVCH - CONTAINS THE PREVIOUS CHARACTER
        STEND - LAST USED STRING DELIMITER
        STRNG - ORIGINAL COMMAND STRING
```

Table 2.2. DECODE VARIABLES.

4. STANDARD APPLICATION COMMONS

Several labeled common areas have been defined to contain information which insures that the code is as machine independent as possible. Programmers should use the variables in common instead of recalculating the values. The programmer should review all common areas to understand their application and contents.

The basic commons that the programmer should review are as follows:

1) Blank Common (unlabeled) - contains all general system information.

2) DEBUGC Common - Contains the debug switch.

3) EDITC Common - Contains all general variables related to editing a document.

4) FDITC Common - Contains all general variables related to accessing a second document while editing a document.

5) IOC Common - Contains all input and output logical device numbers.

6) LOCKC Common - Contains all resource allocation information.

7) MC Common - Contains several machine-dependent constants derived from NCU.

8) PARSC Common - Contains all information about the command being processed.

9) SCRTC Common - Contains one array 243 words long that can be used for table-handler access rather than dimensioning a new local array.

10) SIZEC Common - Contains all information about the number of standard units required to represent specific length character strings.

11) SYSTM Common - Contains all general information about the EDITSPEC system and system table.

12) TABLC Common - Contains information used by the table-handler system.

13) TPXXXX Commons - XXXX is the name of a system or document table. Contains offset information to access the tables and data records.

14) VFMT Common - Contains variable formats for printing different types of character strings.

A listing of all commons used in the EDITSPEC system is given in Appendix E.

5. STANDARD APPLICATION SUBROUTINES

Several basic application programs are available for the programmer's use. To insure uniformity in programming and easy modifications to basic functions, all programmers must use the standard application subroutines provided.

Two lists are provided for reference. The first (Table 2.3) contains the functions to be performed, followed by the name of the subprogram that performs the function. The second (Table 2.4) contains the calling sequence and the definitions of the parameters to be used.

FUNCTION	SUBROUTINE
Character Conversions	
EDITSPEC to Machine	
Packed Strings	ED2MC RCNVR
Single Character	IN2EX
Machine to EDITSPEC	
A1 Format String	MC2ED
Single Character	EX2IN
EDITSPEC to Integer Number	
Within Packed Strings	INTER
Beginning of Packed Strings	INTGR
Integer Number to EDITSPEC	
Packed	NO2IC
Copy Characters to Another String	
From left to right	ICOPY
From right to left	ICOPB
Data Set Number	
Primary	FDSMN
Backup	FDSBK
EDIT A Second Document While EDITING a Primary Document	FDIT FSTOR
Move Characters Within a String	CHIFT
Packing Characters	
EDITSPEC to EDITSPEC	PACK
Unpacking Characters	
Machine to Machine	
packed to A1 format	AN2A1
EDITSPEC to EDITSPEC	
packed to unpacked	UNPAK

Table 2-3. Programming Functions Supported by Standard Subroutines

SUBROUTINE AND VARIABLES		I/O	TYPE	CHANGED
SUBROUTINE:	AN2A1 (SOURC,LNGTH,DESTN)			
FUNCTION:	TO CONVERT FROM (AN) FORMAT; (N=NCU).			
VARIABLES:	SOURC () IS SOURCE STRING PACKED IN A4 FORMAT (FOR IBM360).	INPUT	A(t)	NO
	LNGTH IS NUMBER OF CHARACTERS IN SOURC.	INPUT	WORD	NO
	DESTN() IS DESTINATION ARRAY, WHERE STRING IS PUT IN A1 FORMAT.	OUTPUT	A(t)	YES
SUBROUTINE:	CHIFT (LA,NSHFT,NSHS, NSHE,LP)			
FUNCTION:	SHIFT CHARACTERS TO EITHER THE LEFT OR THE RIGHT OR TO PAD WITH A SPECIFIED CHARACTER.			
VARIABLES:	CALLING SEQUENCE: CALL CHIFT (LA,NSHFT,NSHS,NSHE, LP)			
	WHERE: LA - THE ARRAY IN WHICH SHIFTING IS TO OCCUR. (MAX DIMENSIONS = 126 WORDS)	INPUT/ OUTPUT	A(126)	YES
	NSHFT - THE AMOUNT OF SHIFT IN NO. OF CHARS. (-LT, +RT) VALUE OF 0 WILL BLANK AREA WITH EXT. BLANKS.	INPUT	WORD	NO
	NSHS - LEFTMOST CHAR. POSITION OF AREA TO BE SHIFTED.	INPUT	WORD	NO
	NSHE - RIGHT-MOST CHAR. POSITION OF AREA TO BE SHIFTED.	INPUT	WORD	NO
	LP - THE PADDING OR BLANKING CHARACTER.	INPUT	WORD	NO
SUBROUTINE:	ED2MC(EDARR,SC,NC, UCSET,MCARR,NW)			
FUNCTION:	TO CONVERT A STRING FROM EDITSPEC FORMAT TO MACHINE (A1) FORMAT.			
VARIABLES:	EDARR IS THE SOURCE STRING IN EDITSPEC FORMAT.	INPUT	A(t)	NO

Table 2.4. Standard Application Subroutines

SUBROUTINE AND VARIABLES	I/O	TYPE	CHANGED
SC IS THE STARTING CHARACTER NUMBER OF EDARR TO START CONVRTING.	INPUT	WORD	NO
NC IS NUMBER OF CHARACTERS OF EDARR TO CONVERT.	INPUT	WORD	NO
UCSET IS OUTPUT CHARACTER SET CODE.	INPUT	WORD	NO
MCARR IS OUTPUT STRING IN IN MACHINE (A1) FORMAT.	OUTPUT	A(t)	YES
NW IS NUMBER OF WORDS IN MCARR THAT HAVE BEEN CONVERTED.	OUTPUT	WORD	YES
SUBROUTINE: FUNCTION EX2IN(IXCAR)			
FUNCTION: CONVERTS ONE CHARACTER AT A TIME FROM MACHINE INTERNAL CODE TO EDITSPEC INTERNAL CODE.			
VARIABLES: EXTRN - AN ARRAY WHICH IS DEFINED N ELEMENTS LONG WHERE N EQUALS THE NO. OF POSSIBLE INTERNAL CHARACTER REPRESENTATION CODES ON A PARTICULAR MACHINE.			
IXCAR - IS THE MACHINE CODE VALUE FOR A PARTICULAR CHARACTER.	INPUT	WORD	NOT CHANGED
SUBROUTINE: FDIT(JDOCN,JREAD)			
FUNCTION: TO SET UP A SECONDARY EDIT DOCUMENT.			
VARIABLES: JDOCN(3) IS SECONDARY DOCUMENT NAME.	INPUT	A(3)	NO
IDAC IS DOCUMENT ACCESS CODE RETURNED BY ROUTINE FDAC.	OUTPUT	WORD	YES
SUBROUTINE: FDSBK(DSNM,DSNO)			
FUNCTION: TO GET DATASET NUMBER FROM DATASET NAME (BACKUP - NOT MAIN)			
VARIABLES: DSNM(2) IS 6 CHARACTER DATASET NAME.	INPUT	A(2)	NO
DSNO IS DATASET NUMBER (FORTRAN LOGICAL I/O UNIT NUMBER).	OUTPUT	WORD	YES

SUBROUTINE AND VARIABLES		I/O	TYPE	CHANGED
	TEMPO(8) IS TEMPORARY ARRAY TO STORE UNPACKED NAME. DSERR IS USED TO INDICATE AN ERROR CODE.			
SUBROUTINE:	FDSMN(DSNM,DSNO)			
FUNCTION:	TO GET DATA SET NUMBER FROM DATASET NAME (MAIN - NOT BACKUP)			
VARIABLES:	DSNM(2) IS 6 CHARACTER DATASET NAME.	INPUT	A(2)	NO
	DSNO IS DATSET NUMBER (FORTRAN LOGICAL I/O UNIT NUMBER).	OUTPUT	WORD	YES
SUBROUTINE:	FSTOR			
FUNCTION:	TO TERMINATE ACCESS TO A SECONDARY DOCUMENT.			
SUBROUTINE:	ICOPB(SOURC,SFST,NUM, DEST,DFST)			
	C - COPIES NUM CHARACTERS FROM SOURC (STARTING AT SFST) TO C - DEST (STARTING WITH DFST) BACKWARDS C - ALL OTHER CHARACTERS OF DEST ARE UNCHANGED			
VARIABLES:	SOURCE - PACKED INPUT STRING	INPUT	A(+)	NO
	SFST - RIGHT-MOST CHARACTER TO MOVE	INPUT	WORD	NO
	NUM - TOTAL NUMBER OF CHARACTERS TO MOVE	INPUT	WORD	NO
	DEST - PACKED OUTPUT STRING	INPUT/ OUTPUT	A(+)	YES
	DFST - RIGHT-MOST CHARACTER TO START	INPUT	WORD	NO
SUBROUTINE:	ICOPY(SOURC,SFST,NUM, DEST,DFST)			
	C - COPIES NUM CHARACTERS FROM SOURC (STARTING AT SFST) TO C - DEST (STARTING WITH DFST) C - ALL OTHER CHARACTERS OF DEST ARE UNCHANGED			

SUBROUTINE AND VARIABLES		I/O	TYPE	CHANGED
VARIABLES:	SOURCE - PACKED INPUT STRING	INPUT	A(+)	NO
	SFST - RIGHT-MOST CHARACTER TO MOVE	INPUT	WORD	NO
	NUM - TOTAL NUMBER OF CHARACTER TO MOVE	INPUT	WORD	NO
	DEST - PACKED OUTPUT STRING	INPUT/ OUTPUT	A(+)	YES
	DFST - RIGHT-MOST CHARACTER TO START	INPUT	WORD	NO
SUBROUTINE:	INTEGER FUNCTION INTER(STRNG,IPTR,LENG)			
FUNCTION:	THIS ROUTINE COPIES A NUMERIC STRING FROM (STRNG) AND PLACES IT INTO THE (DIGIT) ARRAY. THE (DIGIT) ARRAY IS THEN CONVERTED INTO A NUMBER.			
VARIABLES:	CALLING SEQUENCE:			
	STRNG = THE STRING OF CHARACTERS REPRESENTING THE USERS INPUT COMMAND.	INPUT	A(+)	NO
	IPTR = FIRST CHARACTER IN THE STRING FOR THE FIRST NUMBER.	INPUT	WORD	NO
	LENG = TOTAL NUMBER OF CHARACTERS IN THE NUMBER.	INPUT	WORD	NO
SUBROUTINE:	INTEGER FUNCTION INTGR (A,NDIG)			
FUNCTION:	THIS ROUTINE FORMS THE ACTUAL NUMBER FOR THE STRING INTO THE MACHINE INTERNAL REPRESENTATION OF THE ACTUAL NUMERICAL VALUE BEING REPRESENTED.			
VARIABLES:	CALLING SEQUENCE:			
	A = NUMERIC STRING	INPUT	A(+)	NO
	NDIG = NUMBER OF DIGITS TO CONVERT	INPUT	WORD	NO
SUBROUTINE:	INTEGER FUNCTION IN2EX(ICAR)			
FUNCTION:	CONVERTS THE EDITSPEC REPRESENTATION OF CHARACTERS INTO THE STANDARD MACHINE CODE REPRESENTATION.			

SUBROUTINE AND VARIABLES		I/O	TYPE	CHANGED
	ICAR IS A WORD WHICH CONTAINS ONLY ONE CHARACTER RIGHT-JUSTIFIED			
	EXTRN IS THE MACHINE'S REPRESENTATION OF CHARACTERS THE EDITSPEC REPRESENTATION IS A CODE FROM 1 TO 62 WHICH CORRESPONDS EXACTLY TO THE ORDERING OF EXTRN			
SUBROUTINE	NO2IC(NUMBR,CHARS,NCHAR)			
FUNCTION:	TO CONVERT AN INTEGER NUMBER TO FOI INTERNAL CHARACTERS. RESTRICTED TO 10 SIGNIFICANT DIGITS.			
VARIABLES:	CALLING SEQUENCE: CALL NO2IC(NUMBR,CHARS,NCHAR)			
	NUMBR IS THE INTEGER NUMBER PASSED TO NO2IC	INPUT	WORD	NO
	CHARS(3) IS AN ARRAY CONTAINING THE CHARACTERS AND IS FILLED AND RETURNED BY NO2IC.	OUTPUT	A(3)	YES
	NCHAR IS RETURNED BY NO2IC AS THE TOTAL NUMBER OF CHARACTERS STORED IN CHARS BY NO2IC.	OUTPUT	WORD	YES
SUBROUTINE:	PACK(SOURC,L1,NUM, DEST,L2)			
FUNCTION:	PACK NUM CHARACTERS INTO THE STRING DEST, STARTING WITH CHARACTER L2 OF DEST. THE CHARACTERS ARE TAKEN FROM THE LEAST SIGNIFICANT CHARACTER OF THE ELEMENTS OF THE ARRAY SOURC, STARTING WITH STANDARD UNIT L1.			
VARIABLES:	SOURC - PACKED INPUT STRING	INPUT	A(+)	NO
	L1 - RIGHT-MOST CHAR TO MOVE	INPUT	WORD	NO
	NUM - TOTAL NUMBER OF CHARACTERS TO MOVE	INPUT	WORD	NO
	DEST - PACKED OUTPUT STRING	INPUT/ OUTPUT	A(+)	YES
	L2 - FIRST CHARACTER IN OUTPUT	INPUT	WORD	NO

SUBROUTINE AND VARIABLES		I/O	TYPE	CHANGED
SUBROUTINE:	RCNVR (INREP, EXREP, NCHAR)			
FUNCTION:	C - CONVERTS THE EDITSPEC INTERNAL REPRESENTATION OF CHARACTERS IN 'INREP' INTO THE MACHINE INTERNAL REPRESENTATION IN 'EXREP'. 'NCHAR' CHARACTERS ARE CONVERTED.			
VARIABLES:	INREP - EDITSPEC PACKED ARRAY	INPUT	A(+)	NO
	EXTREP - MACHINE PACKED ARRAY	OUTPUT	A(+)	YES
	NCHAR - NO. OF MACHINE CHARACTERS	INPUT/ OUTPUT	WORD	YES
SUBROUTINE:	UNPAK (SOURC,L1,NUM, DEST)			
FUNCTION:	UNPACK NUM CHARACTERS FROM THE STRING SOURC, STARTING WITH CHARACTER L1 OF SOURC. THE CHARACTERS ARE PLACED INTO THE LEAST SIGNIFICANT CHARACTER OF THE ELEMENTS OF THE ARRAY DEST. THE REMAINING CHARACTERS OF EACH ELEMENT OF DEST ARE FILLED WITH THE CONTENTS OF THE VARIABLE 'FILL' (ZEROES IN THIS IMPLEMENTATION).			
VARIABLES:	SOURC - PACKED INPUT STRING	INPUT	A(+)	NO
	L1 - STARTING CHARACTER TO UNPACK	INPUT	WORD	NO
	NUM - TOTAL NUMBER OF CHARACTERS TO UNPACK	INPUT	WORD	NO
	DEST - OUTPUT ARRAY	OUTPUT	A(+)	YES

6. STANDARD ACCESS TO TABLES

During the actual execution of EDITSPEC in an interactive or multiprogramming environment, several users may be executing different copies of the EDITSPEC code at the same time. Some of the users may wish to access (to read and/or write) the same resources (document or system table) at the same time. The "multi-user" feature will permit concurrently executing EDITSPEC programs to share the use of system tables and documents. Shared use of these resources must be strictly controlled in order to ensure that one program does not interfere with the correct execution of others. This control takes the form of synchronizing use of these resources on the part of the programs involved.

When a program requires use of a resource, it must request control of that resource from the operating system. Control of a resource can be either exclusive or shared. Exclusive control of a resource guarantees that no other program will be granted access to that resource (write access). Shared control guarantees that no other program will be granted exclusive control of that resource, but other programs will be granted shared control of that resource (read only access).

A request for either kind of control of a resource can be either conditional or unconditional. For conditional requests, control is granted only if the resource is immediately available. The requesting program is informed as to whether or not control was granted. For unconditional requests, control is granted as soon as the resource becomes available. The requesting program may have to wait for an indefinite amount of time. This eventuality is entirely transparent to the program itself.

When a program has finished using a resource, it must relinquish control of that resource so that it becomes available for use by other programs.

REQUESTING AND RELINQUISHING CONTROL OF RESOURCES IS THE PROGRAMMER'S RESPONSIBILITY. Two subroutines, LOCK and UNLOK, have been made available for this purpose.

Required Programming Before Calling LOCK or UNLOK

```
INTEGER RSRCS
COMMON /LOCKC/ RSRCS (7,40), NRSRCS
```

Before calling LOCK or UNLOK, you must describe the resources to be requested or relinquished via the array RSRCS. Each column corresponds to one resource. The number of resources described in RSRCS must be defined in the variable NRSRCS.

Before a call to LOCK, RSRCS(2,I), RSRCS(3,I), RSRCS(4,I), and RSRCS(5,I) must contain the name of the I'th resource requested. RSRCS(1,I) must contain the number of the data set on which this resource resides. RSRCS(7,I) should = 1 if exclusive control of this resource is requested, 0 if shared control is requested. NRSRCS should contain the number of resources. RSRCS(6,I) should not be used within a processing program.

LOCK, as its name implies, is to be used to request control of resources.

LOCK has a single argument, the integer variable IND, which should = 1 if the request is conditional, 0 if the request is unconditional. If IND = 1, LOCK will return the status of the request via the argument IND. IND = 1 if control of all requested resources was granted, 0 if control over none of the resources was granted due to the non-availability of one or more of the resources. If the request was unconditional, then control of all requested resources was granted (IND was not modified and still = 0). Note that LOCK (or UNLOK) never modifies LOCKC.

UNLOK is to be used to relinquish control of resources. UNLOK has no arguments, since control of resources is always relinquished unconditionally.

Application Rules

The following rules must be strictly observed:

(1) Control of a resource should not be requested until it is needed;

(2) A resource must not be used until after its control has been requested and granted;

(3) a. The use of DKOPN, DKPUT, DKINS, DKLOS, DKSET, DKRNM, DKXIT, or DKCLR in connection with a resource requires EXCLUSIVE CONTROL of that resource;

b. The use of DKFIL, DKLEN, or DKGET in connection with a resource requires shared or exclusive control of that resource;

c. The use of DKNDS, DKNIT, or DKDMP is unrestricted;

(4) Control of a resource must not be relinquished until it is consistent with respect to both itself and other resources;

(5) Control of a resource should be relinquished as soon as it is no longer needed;

(6) A resource must not be used after its control has been relinquished;

(7) The logic of execution should be such that: UNLOK is not called without a previous call to LOCK; two calls to LOCK/UNLOK are never made without an intervening call to UNLOK/LOCK; LOCK is not called without a subsequent call to UNLOK;

(8) a. The same resource must not be referenced by two different columns in RSRCS;

b. The maximum number of resources to be locked is 40;

(9) No assumptions may be made regarding the contents of a resource at the time its control is granted;

Failure to observe one or more of the above rules will not necessarily result in the occurrence of a perceptible error condition (such as an ABEND), but resource integrity may nevertheless be seriously impaired. It is most important that great care in coding be exercised.

Obtaining Access From System Commands

When CMMND transfers control to a system command, no resources have been locked and the programmer should start from the first column to request resources. The programmer should obtain access to all resources needed at the beginning and give up all resources at the end of the system subroutine. NRSRCS should always be set to zero before returning from a system command.

Obtaining Access From EDIT and Internal Commands

When CMMND transfers control to an edit or internal command, access to the resources for the document tables have already been obtained by the EDIT or FDIT subprograms which were previously called.

Access to system tables must be performed by unlocking all resources, adding the system table to the end of the current list, and locking all resources. Removing access to system tables must be done by unlocking all resources, removing the last system tables that were locked, and locking the remaining resources if there are any in the array.

7. ADDING NEW FUNCTIONAL COMMANDS

Addition of the new functional commands should be performed in the following manner:

1. Write Users Manual Entries - Review the format for user commands as given in the Users Manual. The command description to be entered into the Users Manual should be prepared from the functional requirements provided by the proponent agency. Typing instructions are in Appendix A.

2. Review Users Manual Entries - The new text should be reviewed and approved by the proponent agency before any detailed design or coding is started.

3. Prepare SIT Test Deck - The complete test deck for each new command should be written. See Appendix B for an example.

4. Review SIT Test Deck - The test deck should be reviewed and approved by the proponent agency.

5. Design Subroutines - Prepare a description of the design in correct English prose. Prepare this description on coding sheets as comments to be used for documentation in the actual code.

6. Code Each Subprogram - The programmer should set up the standard subprogram documentation forms prior to coding the first line of code. A detailed description of the design in correct English prose should be prepared on coding sheet before actual coding is started. Code and document all coding as the coding is performed.

Subroutines CMMND and MATCH must be modified to transfer control to the correct processing program. See Appendix C for a Documentation Example.

See Appendix D for Programming Notes and Procedures.

7. Set up standard debug options for each subprogram.

The command .DEBUG X. will work in both system and edit mode to turn on the debug trace variable DEBUG located in /DEBUGC/ common. Every program should be written as follows with an IF statement to check the value of DEBUG:

X = 0 no debug print of any type
X = 1 The message "IN <SUBROUTINE NAME>" printed with the values of all variables passed to the routine in the subroutine call and in common. This is the first executable statement in the subroutine. If a common variable may be changed, it should be printed here first.

The message "OUT <SUBROUTINE NAME>" printed with the same variables as the "IN" write statement. This is the last executable statement in the subroutine.

X = 2 Detailed debug output

All debug statements should be labeled as "GUBED-C" as the last seven characters in card column 74 through 80. This automatically identifies all debug statements.

Debug statements without the "IF" check are permissible for debugging purposes. These cards should never be removed from the program. Instead, the cards should be reversed to show the C-DEBUG as the first characters. This will allow another programmer to reverse the cards again to obtain the real detailed debug print.

If this procedure is followed, no debug work will ever be lost and have to be redone by a future programmer.

The list routines .LIST. and .LT. are provided to allow the programmer to adequately test his programs. If new tables are added to the system, new list routines should be rewritten for each new table.

8. Hand Debug the SIT Deck - The programmers should play computer and write out a complete trace of each and every command in the SIT deck. This should insure that the complete logic of the program has been tested and the program is correct.

9. Computer Debug the SIT Deck - The programmer should run the SIT deck with the DEBUG switch set to 2 and compare the results with the hand debug output.

10. Complete Cross-Reference Indices - There are several indices that need updating:

1. Commons used
2. Tables used
3. Subprogram cross-reference (called & called by) index cards
4. Conversion forms

APPENDIX A

USER MANUAL TYPING NOTES

The text to be typed is a portion of the "EDITSPEC USER'S MANUAL." The manual is composed of several hundred "COMMANDS" and "CHAPTERS."

Each command and chapter should be typed and stored as a separate document (or file).

Each command and chapter document (or file) should be named "EDITSPEC USER'S MANUAL - command name" or chapter number.

Example - Command Name = INPUT
EDITSPEC USER'S MANUAL = INPUT

All documents will be stored until FY84.

PLEASE TYPE IN FINAL FORM ACCORDING TO ATTACHED INSTRUCTIONS.

USER MANUAL TYPING INSTRUCTIONS

1. A one-inch margin should be placed on all four sides of an 8 by 10 1/2" sheet of paper.

2. Each command will begin on a new page.

3. The complete phrase applied to define the command will be typed in capital letters and right justified on the first text line of every page related to the command.

4. The shortest acceptable character string will be typed in capital letters on the second text line of every page related to the command. The first character will be typed immediately below the first character of the complete phrase applied to define the command.

5. The titles of the subject areas:

TITLE	FIRST CHARACTER INDENTION
PURPOSE	45
GENERAL FORM	42
FIELD OPTIONS	42
SPECIAL NOTES	42
EXECUTION PROCEDURES	39
COMMAND VARIATIONS	39
MESSAGES	43

will be typed in capital letters, centered, and underlined on a new line. Two lines are to be skipped before and after the subject area title. If the subject is not required, the word "None" will be printed after the title.

Example: FIELD OPTIONS - None

6. The "GENERAL FORM" subject area is composed of two different typing formats. The first typing format will be boxed in by an illustrator before printing. An additional three lines will be skipped after the GENERAL FORM title to allow for the box. This is a total of 5 lines to be skipped after the general format title. Two lines after the title and three lines for the box before the first text line. All lines should be indented 5 spaces from each margin. The first few characters, the shortest acceptable character string, will be typed in capital letters. The rest of the text will be typed in italics. Stop codes should be placed to allow type ball changes. Italics should not be used in drafts of the commands. This will speed up the draft preparation. Three lines will be left after the completion of this format for the illustrator's box.

The second typing format contains a three-column table. The first column, 12 characters in width, contains the word "where," left justified in the first row. The first column is blank for all other rows. The text in the second column, 24 characters in width, will be typed left justified and in italics. Stop codes should be placed to allow type ball changes. The text in the third column, 36 characters in width, will be typed left justified in regular type. The first column should have 12 spaces, the second column 24 spaces, and 36 for the third column. The first word should not be capitalized.

7. FIELD OPTIONS. The field options section is composed of a three-column, 24 spaces each, table. The table rows are separated by one skipped line, one straight line, and one skipped line. The column titles "FIELD, OPTIONS, DEFAULT" should appear on each page that contains this section. Text in the first column is in italics, and the rows are numbered by integers with a period and 2 spaces following the integer. The second and third columns are in normal type. The first, second, and third columns should be 24 spaces wide. The first word should not be capitalized. Leave at least 2 spaces between columns.

8. SPECIAL NOTES. Use italics where noted.

9. EXECUTION PROCEDURES. Use italics where noted.

10. COMMAND VARIATIONS. Each variation may contain the following formats: The first is the variation number, left justified, marked "Var. #n". This is followed by the command in upper case regular, then the Italics type. This line should be underlined. The second is the Example number indented 5 spaces and underlined. The general form of the third format indented 5 spaces and underlined is:

Text before:
Command:
Text after:

11. MESSAGES. If no messages are to be typed the phrase "All messages self-explanatory" should be placed after "MESSAGES."

1"

ENTER
EN

2 Lines

PURPOSE

2 Lines

This edit command allows the user with write access to insert a new line of text into an existing document.

2 Lines

GENERAL FORM

5 Lines

.EN*ter line number; text; text segment id*

1" 1"

3 Lines

where	*line number*	is the 1- to 8-digit line number of the new text to be inserted.
	text	is line of not more than 384 characters, counting capital letters twice. The string must be enclosed in parentheses or a special character.
12 spaces	*text segment* 24 spaces	is a 1- to 4-character alphanumeric id.

1" 2" 3" 36 spaces

2 Lines

FIELD OPTIONS

2 Lines

FIELD	OPTIONS	DEFAULT
1. *line number*	valid line number	add text line to end of document.
2. *text*	1-384 alphanumeric characters, counting capital letters twice	no default. field required.

1 Line
1 Line

2" 24 spaces 2" 2"

1"

FIELD	OPTIONS	DEFAULT
3. *text segment*	1- to 4-character alphanumeric Id.	no text segment assigned.

2 Lines

SPECIAL NOTES - None

2 Lines

EXECUTION PROCEDURE

2 Lines

If the line number field is blank, then the document's current increment is added to the last number in the document to calculate the entered text line number.

2 Lines

COMMAND VARIATIONS

2 Lines

Var. #1. .EN_*line number; (text) text segment*

Example #1.

Text before: Text Table

LN	Text
14000	Additional ----- shingles
14200	*p3*open ------- accordance

Command: .EN_14100; (on each side of the valley);A.

Text after: Text Table

LN	Text
14000	Additional ----- shingles
14100	on each side of the valley.
14200	*p3*open ------- accordance

The text is entered on line 14100 with a text segment id of A.

Var. *2. .EN_ *line number; (text)*

Example #1.

Text before: **Text Table**

LN	Text
15200	two -------- each
15400	*p3* ------- surfaced

Command: EN_15300; (shingle tab along open metal valleys.)

The text will be added as line 15300 with no text segment.

Text after: **Text Table**

LN	Text
15200	two -------- each
15300	shingle tab along open metal valleys.
15400	*pe* ------- surfaced.

Var #3. .EN_; *(text); text segment*

Text before:

LN	Text
22500	construction ---- wind
end	

Command: .EN_ (areas ,*s1*).

The new line will be added after line 22500 with the documents increment (100).

Text after: **Text Table**

LN	Text Table
22500	construction ---- wind
22600	areas, *SL*
end	

↕ 2 Lines

MESSAGES - All messages self-explanatory.

APPENDIX B

SIT TEST DECK EXAMPLE

CERL TEST DECK 2/11/

```
.NEW   PRINTTEST:TESTDATA .
~~.EDIT  PRINTTEST .~~
.INPUT .
  THIS DOCUMENT TESTS THF'PRINT' COMMAND . ALL FIFLD WILL BE TESTED FOR EACH
  SURFIELD DELIMETER AND EACH PARAMETFR TYPE .
 THE PRINT COFMAND MUST INTFRACT WITH THE BEGIN PARAGRAPH AND PAGE NUMBER
 COMMANDS, AND ALL FORMAT TABLES.
  THERE ARE NO INTERRFLATIONSHIPS BETWEEN FIELDS.
   INITIALIZATION  REQUIRED TO TEST PRINT COMMAND
      DEFINE  DOCUMENT FORMAT UNDER ID OF 1 & 22
  FIELD 1 - DOCUMENT FORMAT ID
        TEST NUMBER             DESCRIPTION
          1.  BLANK FIELD -(EX10
          2.  NOT IN THE SYSTEM TABLE-(EX 30
          3.  MORE THAN EIGHT INTEGER CHARACTERS  (EX23
          4.  IN THE SYSTEM TABLE  (EX1,8,9,11 THR
  FIELD 2 -   COLUMNS TO BE PRINTED
          5.  BLANK FIELD (EX8,9,23
          6.  UNALLOABLE LETTER (EX1,10
          7.  LENGTH GREATER THAN ONE CHARACTER (EX2
          8.  F 16, 14.-F (EX12,31
          9.  L 17  15.-L(EX13,32
         10.  T 18  16.-T(EX11,33
         11.  A 21  17.-A(EX12,31
         12.  X 20, 18.-X(EX11
         13.  P 21, 19.-P(EX22
         20.  REPETITION OF THE SAME CHARACTER (EX14
         21.  BOTH THE LETTER AND ITS NEGATIVE (EX14
  FIELD 3 - ARFA
             STARTING  ENDING               TEXT SEGMFNTS
             LINE NO.  LINE NO.    ONE      TWO       THREE
             --------  -------     ---      ---       -----
         22.  BLANK     BLANK      BLANK    BLANK     BLANK (FX8,9
         23.    0         0        BLANK    BLANK     BLANK (FX10,20
         24.    0       GTR SLN    BLANK    BLANK     BLANK (EX11
         25.    0       LES SLN    BLANK    BLANK     BLANK (FX12
         26. GTR ELN      0        BLANK    BLANK     BLANK (EX13
         27. LES ELN      0        BLANK    BLANK     BLANK (EX14
         28.    0         0        VALUE1   VALUE2    VALUE3(EX34
         29.  BLANK     BLANK      VALUE1   VALUE2  (EX16,17
         30.  NUMBER    NUMBER     VALUE1           (EX18
         31.  9 CTR NO  9 CTR NO   VALUE1   VALUE1 (EX23(9),12
         32.                       5 CTR ALP 5CTR ALPHA   (FX13
         33.   ONE OR THREE NOS          FOUR TEXT SEGMENTS(EX24,25,26,27
         34.A  TEXT SEGMENT     LINE NUMBER LIST  (EX35
         34.B NUMBER1   NUMBER1                 (EX1
         34.C                      NUMBER       (EX1
         34.D  ZERO     BLANK                   (EX15,19
  FIELD 4 - LINE SPACING
         35. BLANK    (EX8,9
         36.   0      (EX14
         37. SINGLE-1 (EX1,18
        38. DOUBLE-2 (EX20
        39.   3+ (EX10.11
        40. TWO NUMBERS (EX17.23
   FIELD 5- NO INDEX NOR TABLE CREATION
        41. BLANK  (EX8.9
        42. ZERO(EX14
        43. ONE (EX1.18
        44. TWO + (EX10.11
        45. TWO NUMBERS (EX17.23
   FIELD 6- LOGIC CONDITION OVER RIDE
        46. BLANK  (FX8.9
        47. ZERO (EX14
        48. ONE  (EX1.18
        49. TWO + (EX10.11,23
        50. TWO NUMBERS (EX17.23
   FIELD 7 - OUTPUT DEVICE NUMBER
        51. BLANK  (EX8.9
        52. ZERO   (EX14
        53. ONF  (EX1.18
        54. TWO + (EX10.11
        55. TWO NUMBERS(EX17
   FIELD 8 - PAGE/PARAGRAPH NUMBER INITIALIZATION
        56. BLANK (EX8.9
        57. ZERO (EX14
        58  ONE  (EX1.18
        59  TWO + (EX10.11
        60  TWO NUMBERS (EX17
   S= SYNTAX ERROR
   EXAMPLE 1 - ONE CHARACTER NUMERIC FOR EACH FIELD AND SUBFIELD
               F1-4;F2-6(5);F3-348.C;F4-37;F5-43;F6-48;F7-53;F8-58
 .PR   1;   1, 1, 1, 1. 1;     1-     1,     1.     1.     1;  1 ;  1 ;  1 ;  1 ;   1.
   EXAMPLE 2 - ONE CHARACTER ALPHABETIC FOR EACH FIELD AND SUBFIELD  F2- 7
               F1-S;F2-11(5);F3-S.S;F4-S;F5-S;F6-S;F7-S;F8-S
 .PR   A;  AA, A, A, A, A;     A-     A.     A,     A.     A;  A ;  A ;  A ;  A ;   A.
   EXAMPLE 3 - ONE CHARACTER ALPHABETIC IN DELIMITERS
               F1-S;F2-S;F3-S;;4-S;F5-S;F6-S;F7-S;F8-S
 .PR (A); (A),  ,  ,  ,  ;  (A)-  (A),  (A).  (A),  (A); (A); (A); (A); (A);(A) .
   EXAMPLE 4 - ONE HYPHEN IN EACH FIELD - SYNTAX
 .PR  - ;  - , -,  ,  ,  ;   - -    - ,    - ,    - ,    - ;  - ;  - ;  - ;  - ; -  .
   EXAMPLE 5 - ONE COMMA  IN EACH FIELD - SYNTAX
 .PR  , ;    ,  ,  ,  ,  ;   , -    , ,    , ,    , ,    , ;  , ;  , ;  , ;  , ; ,  .
   EXAMPLE 6 - DOUBLE ASTERIX IN EACH FIELDSYNTAX
 .PR ** ;  **,**,**,**,**; **   -   ** ,   ** ,   ** ,   ** ; ** ; DD ; ** ; ** ; ** .
   EXAMPLE 7 - ONE PLUS IN EACH FIELD
 .PR  + ;  + , +, +, +, +;  +   -    + ,    + ,    + ,    + ;  + ;  + ;  + ;  + ; + .
   EXAMPLE 8 - COMPLETE DEFAULT OPTION WITH DELIMETERS
               F1-1;F2-5;F3-22;F4-35;F5-41;F6-46.F7-51.F8-56
 .PR   1;    ,  ,  ,  ,  ;     -      ,      ,      ,      ;    ;    ;    ;    ;     .
   EXAMPLE 9 - COMPLETE DEFAULT OPTION WITH NO DELIMETERS
               F1-1;F2 THR F8 BLANK
 .PR 1.
   EXAMPLE 10- F1-1;F2-6(3);F3-23;F4-S.39;F5-S.44;F6-S,49;F7-S.54;F8-S.59
 .PR    ;   B, C. D,  ,  ;     0-0    ,      ,      ,      ; -5 ; -5 ; -5 ;  -5; -5 .
   EXAMPLE 11- F1-4;F2-16.18;F3-24;F4-39;F5-44;F6-49;F7-54;F8-59
 .PR  22;  -X,-T,  ,  ,  ;     0-9999 ,      ,      ,      ;  5 ;  5 ;  5 ;   5;  5 .
   EXAMPLE 12- F1-4;F2-14.17;F3-25;F4 THR F8  BLANK    F3-ALSO-31
 .PR  22;  -F,-A,  ,  ,  ;     0-1    ,TSA   .TSA   ,TSA   ;    ;    ;    ;    ;    .
```

```
  EXAMPLE 13- F1-4;F2-15;F3-26;F4 THR F8 BLANK    F3-ALSO-32
.PR  22;  -L,  ,  ,  ,  ; 9999-0    ,ABCDE,     ,     ;     ;     ;     ;     ;     .
  EXAMPLE 14- F1-4;F2-20,21;F3-27;F4-36;F5-42;F6-47;F7-52;F8-57
.PR  22;  -F,-F, F,  ,  ;     1-0     ,     ,     ,     ; 0   ;  0  ;   0;    0;0  .
  EXAMPLE 15- F1-4;F2-8;F3-34D;F4 THR F8 NO PROCESSING
.PR  22;   F,  ,  ,  ,  ;     0-      , TSGO, 1TSG, 2TSG;     ;     ;     ;     ;     .
  EXAMPLE 16- F1-4;F2-8;F3-29;
.PR  22;   F,  ,  ,  ,  ;             , TSGO, 1TSG, 2TSG;     ;     ;     ;     ;     .
  EXAMPLE 17- F1-4;F2-9;F3-29;F4-17;F5-45;F6-50;F7-55;F8-17
.PR  22;   L,  ,  ,  ,  ;     -       , TSGO, 1TSG,     ; 12  ; 12  ; 12  ; 12  ; 12 .
  EXAMPLE 18- F1-4;F2-10;F3-30;F4 THR F8  ONE
.PR  22;   T,  ,  ,  ,  ;     1-9999  , TSGO,     ,     ; 1   ;  1  ;   1;    1;1  .
  EXAMPLE 19- F1-4;F2-11;F3-
.PR  22;   A,  ,  ,  ,  ;      -9999  ,     ,     ,     ;     ;     ;     ;     ;     .
  EXAMPLE 20- F1-4;F2-12,F3-23  F4-38
.PR  22;   X,  ,  ,  ,  ;     0-0     ,     ,     ,     ;  2  ;     ;     ;     ;     .
  EXAMPLE 21- F1-4;F2-13,11
.PR  22; P   ,A ,  ,  ,  ;     -       ,     ,     ,     ;     ;     ;     ;     ;     .
  EXAMPLE 22- F1-4;F2-19,
.PR  22; -P ,  ,  ,  ,  ;     -       ,     ,     ,     ;     ;     ;     ;     ;     .
  EXAMPLE 23- F1-3;F2-5;F3-31;F4 THR F8 -.CTRS
.PR 123456789; ;123456789-123456789;123456789;123456789;123456789;123456789;
  EXAMPLE 24- F1-4;F2-5;F3-33
. 123456789.
.PR  1;;2- 3,4-5.
  EXAMPLE 25- F1-4;F2-5;F3-33
.PR  1;;2-  ,4- .
  EXAMPLE 26- F1-4;F2-5;F3-33
.PR  1;; - 3, -5.
  EXAMPLE 27- F1-4;F2-5;F3-33
.PR  1;; A1,A2,A3,A4,A5.
  EXAMPLE 28  ALL FIELDS MISSING
.PR .
  EXAMPLE 29  TOO MANY FIELDS -ALL BLANK
.PR ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; .
  EXAMPLE 30  F1-2;
.PR 4.
  EXAMPLE 31  F1-4;F2-14
.PR 1;-F,-A.
  EXAMPLE 32  F1-4;F2-15
.PR 1;-L.
  EXAMPLE 33  F1-4;F2-16
.PR 1;-T.
  EXAMPLE 34  F1-4;F25;F3-28
.PR 1;;TSGO,1TSG,2TSG.
  EXAMPLE 35  F1-4;F2-5;F3-34A
.PR 1;;TSGO,1TSG,2TSG,1-999.

.STORE.
.EDIT  PRINTTEST.
.EX ;;1.
.STORE.
```

APPENDIX C

DOCUMENTATION EXAMPLE

DOCUMENTATION EXAMPLE

```
      SUBROUTINE PRINT      (STRNG.PMPTR)
C     PROGRAM NAME: PRINT     (STRNG.PMPTR)
C     FUNCTION: THIS ROUTINE CHECKS ALL INPUT DATA AND PRINTS ALL ERRORS. IT
C               THEN TRANSFERS CONTROL TO (PRMON) TO DO PRINTING.
C     AUTHOR: UNKNOWN AUGUST 1976
C     MODIFICATIONS: NEELY SEPT 76
C     LANGUAGE: FORTRAN
C     CALLING SEQUENCE : STRNG=THE STRING OF CHARACTERS REPRESENTING THE
C                             USERS INPUT COMMAND.
C                        PMPTR=A 2-DIMENSIONAL ARRAY CONTAINING THE
C                             DESCRIPTION AND LOCATION OF ALL PARAMETER
C                             FIELDS WITHIN THE USERS COMMAND-STRNG.
C     ROUTINES CALLED:
C        INTER-FUNCTION TO FORM AN INTEGER.
C        UNPAK
C         LNLIM =       OBTAUNS FIRST AND LAST LINE NUMBERS
C        PRMON        PRINTS TEXT
C          TBLCS-RECORD LOCATE IN TABLE
C          ICOPY-
C     TASKS OR MODULES:
C     VARIABLES:
C        STRNG(1)= USERS COMMAND
C        PMPTR(100.3)= 99 PARAMETER FIELDS.3 COLUMNS-1ST CTR.LENGTH.TYPE
C        NPRI=NO OF THE DEVICE TO PLACE THE PRINTED DOCUMENT ON
C        FLGOP(6)= DATA COLUMNS TO BE PRINTED: 1- X-50    .2- 38-L
C                  3-32-F              4-46-T              5-27-A  6-42-P
C        IMERR= 1-GENERAL ERROR MESSAGE PRINTED. 0-NOT PRINTED FOR ONE FIELD.
C        IERR = 0-NO ERRORS. 1- ERRORS. STOP AFTER PARAMETER ANALYSIS
C        MODE = THE FIELD NO. BEING PROCESSED. NUMBERED IN ORDER 1 TO 8
C        IHYPH= 0=HYPHEN NOT PREVIOUS DELIM. 1= HYPHEN IS PREV DELIM.
C        IELN= ENDING LINE NUMBER TO PRINT.USER INPUT
C        ISLN= STARTING LINE NO   TO PRINT.USER INPUT
C        ISEG=      TEXT SEGMENT 1 THR3
C    EDITF- TABLE BEING EDITED
C        ISPAC= LINE SPACING01=SINGLE 2= DOUBLE
C        ICNT = NO. OF PARMS PLUS ONE -ROW LOCATION OF LAST PARAMETER
C        ITYPE= PARM. TYPE
C        IPTR = 1ST CTR OF PARM. IN STRNG ARRAY
C        LENG = PARM. CTR LENGTH
C        DEBUG= 2 OR GTR FOR DEBUG TESTING
C        FLGNM- 1-50=X  .2-38=L  .3-32=F .4-46=T .5-27= A   ALL AVAILABLE FIELDS
C               6-31=E
C        LETTER = THE COLUMN PRINT OPTION USER INPUT VALUE
C        JSLN= STARTING LINE NO. IN TEXT TABLE.
C        JELN= ENDING LINE NO. IN TEXT TABLE.
C        ISET= 1=PRINT  .-1= NO PRINT
C        I = THE ROW IN PMPTR BEING ANALIZED.
C        J          COUNTER
C     NPPIN- NO P/P INITIALIZATION
C        NOD-NUMBER OF OUTPUT DEVICE 0 OR 1
C        NPRF- ACTUAL PRINT DEVICE
```

```
C         LCO/ LOGIC CHECK OVERRIDE
C         NITC/ NO INDEX/TABLE CONTENTS
C           IRELN- ELN REC ID
C         IRSLN- SLN REC ID
C          IRDFI- DOC FMT ID REC ID
C         FILMN- FILE NAME FOR DATA HANDLER
C         SYSFL- SYSTEM FILE NAME
C         ISDOC- DOC FORMAT TABLE WORD AND FIRST REC ID
C         IS = TEXT SEG COUNTER  1-3 ARE VALID
          JBLNK = BLANK WORD -EDITSPEC          EDTBL
          ISET = PRINT COLUMN INDICATOR  1=YES, -1 = NO
C         ISHID = SUPER HEADER RECORD ID
C         IHDR  = HEADER RECORD
C         ITEM  = (+)LOCATION OF ITEM IN HEADER ,(-)NOT FOUND,WOULD
C                     APPEAR BEFORE THIS ITEM
C         LOC   = LOCATION OF RECORD ID IN HEADER.
C         LNP LINE NUMBERS PROCESSED
C           IFORM= DOCUMENT FORMAT ID
C           MESG = MESSAGE DEVICE NUMBER
C           DFDRC(48)= DOCUMENT FORMAT DATA RECORD
C           I1 = NUMBER OF CHARACTERS IN DOC FMT DATA RECORD
C           NC = NUMBER OF CHARACTERS READ FROM DKGET
C         NPRT=  UNIT NUMBER FOR PRINT DOC ON THE TERMINAL
          NPRP=  UNIT NUMBER FOR PRINT DOC ON HIGH SPEED PTR
C        J = LOCATION OF THE NEXT PARAMETER WHICH SHOULD BE A HYPHEN.
C-      PROGRAM LOGIC:
C
C
C       BLANK COMMON AND EDITSPEC EQUIVALENCES
        INTEGER FILNM(4)
        INTEGER RVRSE,TRU,FALS,ZEROS,ONES,FOLID,PREID,FSTAT,P
        COMMON P(200)
        EQUIVALENCE (P(5),NBW),(P(6),NCW),(P(7),NAW),(P(8),NBC),(P(9),NCU)
        EQUIVALENCE (P(10),NBU),(P(11),NAU),(P(12),NBU2),(P(13),NC2U)
        EQUIVALENCE (P(14),RVRSE),(P(15),NWPSU),(P(16),LFIW),(P(17),LFIWU)
        EQUIVALENCE (P(24),ICRD),(P(25),IPRT),(P(26),IPCH),(P(27),ITAP)
        EQUIVALENCE (P(38),TRU),(P(39),FALS),(P(40),ZEROS),(P(41),ONES)
        EQUIVALENCE (P(187),MAXGP),(P(188),NDS),(P(189),FILNM(1))
        EQUIVALENCE (P(193),FOLID),(P(194),PREID),(P(195),LRLEN)
        EQUIVALENCE (P(196),FSTAT),(P(197),NDA),(P(198),IDHTR)
        EQUIVALENCE (P(199),IDHER),(P(200),IDKER)
C
C
C
C       SYSTEM COMMON
        INTEGER SDS,SBDS,SYSFL,DEBUG,BATCH,USID,EDTBL
        LOGICAL MULTI,BAKUP
        COMMON /SYSTM/ IBLNK,JBLNK,SDS,SBDS,SYSFL(4),USID(3),LOGGD,MULTI
       1BAKUP,BATCH,DEBUG,ISQTY(5,13)
        DIMENSION ISACS(5),ISCHA(5),ISDIR(5),ISSPC(5),ISUSR(5),
       1ISDOC(5),ISFOF(5),ISHEF(5),ISPGN(5),ISPRF(5),ISPRN(5),ISTTF(5),
       2ISBAK(5)
        EQUIVALENCE (ISACS(1),ISQTY(1, 1)),(ISCHA(1),ISQTY(1, 2))
        EQUIVALENCE (ISDIR(1),ISQTY(1, 3)),(ISSPC(1),ISQTY(1, 4))
        EQUIVALENCE (ISUSR(1),ISQTY(1, 5)),(ISDOC(1),ISQTY(1, 6))
        EQUIVALENCE (ISFOF(1),ISQTY(1, 7)),(ISHEF(1),ISQTY(1, 8))
        EQUIVALENCE (ISPGN(1),ISQTY(1, 9)),(ISPRF(1),ISQTY(1,10))
```

```
      EQUIVALENCE (ISPRN(1),ISOTY(1,11)),(ISTTF(1),ISOTY(1,12))
      EQUIVALENCE (ISBAK(1),ISOTY(1,13))
      EQUIVALENCE (EDTBL,IBLNK),(MCHBL,JBLNK)
C
C
C
C     DOCUMENT EDIT COMMON
      INTEGER EDITF,EDS,EBDS
      LOGICAL EREAD
      COMMON /EDITC/ EDITF(4),EDS,EBDS,EREAD,IEDIT,MGO,ICYC,IDAC,INC,
     1               ITOTY(5,15),IERR
      DIMENSION ITAUG(5),ITAUO(5),ITAUT(5),ITDRT(5),ITERT(5),
     1          ITERC(5),ITERL(5),ITERS(5),ITFLG(5),ITIX (5),
     2          ITLC (5),ITPUL(5),ITABC(5),ITEXT(5),ITBAK(5)
      EQUIVALENCE (ITAUG(1),ITOTY(1, 1)),(ITAUO(1),ITOTY(1, 2))
      EQUIVALENCE (ITAUT(1),ITOTY(1, 3)),(ITDRT(1),ITOTY(1, 4))
      EQUIVALENCE (ITERT(1),ITOTY(1, 5)),(ITERC(1),ITOTY(1, 6))
      EQUIVALENCE (ITERL(1),ITOTY(1, 7)),(ITERS(1),ITOTY(1, 8))
      EQUIVALENCE (ITFLG(1),ITOTY(1, 9)),(ITIX (1),ITOTY(1,10))
      EQUIVALENCE (ITLC (1),ITOTY(1,11)),(ITPUL(1),ITOTY(1,12))
      EQUIVALENCE (ITABC(1),ITOTY(1,13)),(ITEXT(1),ITOTY(1,14))
      EQUIVALENCE (ITBAK(1),ITOTY(1,15))
C
C
C
C     I/O UNIT NUMBERS
      COMMON /IOC/ INP1,INP2,NLST,NPRO,NECO,NPRI,MESG ,NPRT,NPRP
C
C
C
C     READ AND PARSE COMMAND ROUTINES COMMON AREA.
      INTEGER EBUFF,PMPTR,STRNG
      LOGICAL EOFIL,NXTRD,CMDOK
      COMMON /PARSC/EBUFF(400),STRNG(400),PMPTR(100,3),NXTLN(80),
     1              NCHAR,ISUB,EOFIL,NXTRD,CMDOK
C
C
C
C     AUXILIARY MACHINE SPECIFIC CONSTANTS
      COMMON /MC/ MAXCH,IWD1,NCUM,IWD2,NCUM2,NCU2,IWD3
C
C
C
C     TABLE HANDLING ROUTINES COMMON AREA
      COMMON /TABLC/ NLOC,ISHDR(11),IRHDR(243)
C
C
C
C     PRINT ROUTINES COMMON AREA
      COMMON /PRINT/ NITC,LCO,NOD,NPPNI
     1  , DFDRC(48),IRSLN,IRELN,ISPAC,ISEG(3)
C
      INTEGER EDITF      ,DFDRC
       DIMENSION  IHDR(243)
      INTEGER FLGNM(6),FLGOP(6)
     1,STRNG(1),PMPTR(100,3)
      DATA FLGNM/50,38,32,46,27,42/
```

```
C-----PUT PRINTED DOCUMENT OUT ON FT08F001 OR FT07 HIGH SPEED PTR
C ••• THE FOLLOWING 10 CARDS ARE FOR DEBUG ONLY
          IF(DEBUG .NE. 2 ) GO TO  9994
          I2= PMPTR(1.3) •1
          I1= (PMPTR(I2.1)• PMPTR(I2.2) / NCU • 1
          WRITE(MESG.9996) (STRNG(I).I=1.I1)
 9996     FORMAT(1H .12Z10 )
          WRITE( MESG.9995)
    1         ((PMPTR(I.J).J=1.3).I=1.I2)
 9995     FORMAT (1H .30I4 )
 9994     CONTINUE
C-     INITIAL ALL VARIABLES TO THE DEFAULT OPTIONS.
         LNP=0
        IS=0                                                                       1
        NITC = 0
        LCO  = 0                                                                   1
        NOD  = 0
         NPRI=NPRT
        NPPIN= 0                                                                   1
        DO  4    I= 1.3
    4    ISEG(I)= EDTBL
C-     MARK COLUMNS TO BE PRINTED AS ZERO FOR NO DECISSION MADE.
       DO 3 I=1.6
       FLGOP(I)=0
    3  CONTINUE
C      SET TO -1 TO INDICATE NO USER ENTRY
        IELN= -1
        ISLN= -1
       IMERR=0
       IERR=0
       MODE=1
       IHYPH=0
       IFORM=0
       ISPAC=1
C-     CHECK FOR NO PARAMETERS IN COMMAND. GTR THAN 0.LESS THAN 35
       ICNT = PMPTR(1.3)•1
        IF(ICNT .LE.  1)GO TO 991                                                  1
        IF(ICNT .GT. 37)GO TO 992                                                  1
C
&

C
C
C-     PROCESS EACH PARAMETER IN SEQUENCE. CHECK FOR ALL ERRORS.
C
C

C

    5 DO 500 I=2.ICNT
        ITYPE=PMPTR(I.3)                                                           1
        IPTR =PMPTR(I.1)                                                           1
        LENG =PMPTR(I.2)                                                           1
        IF(DEBUG .EQ.  2 ) WRITE(6.9000) I.ITYPE.IPTR.LENG                         1
 9000   FORMAT(20H I.ITYPE.IPTR.LENG =. 4I5  )                                     1
C
C
```

Note use of Blank Comment Cards

```
C
C
        GO TO (10,20,900,40,50,60,900,900),ITYPE ← Transfer on
&                                                  Field Type 1 thr8
C
C              C 1= Numeric
C
C
C-    PROCESS NUMBERS.
  10  GO TO (150,900,100,180,130,140,160,170),MODE ← Transfer on                1
&
C                                                  Fieldnumber in
C      DOCUMENT FORMAT ID
  150      IF (LENG  .LE. 8 ) GO TO 959              actual command              1
           WRITE(MESG,9959)                                                      1
 9959      FORMAT (5X,45H*** DOCUMENT FORMAT GREATER THAN 8 CHARACTERS    )      1
           IERR=1                                                                1
        GO TO 500
  959      CONTINUE                                                              1
        IFORM = INTER(STRNG,IPTR,LENG)
C LOAD THE DOCUMENT FORMAT TABLE INTO THE DATA HANDLER FILE LOCATION
  600   DO 601    J=1,4
  601   FILNM(J) = ISDOC(J)
        ISHID    = ISDOC(5)
        CALL TBLCS( ISHID,IFORM,IHDR,ITEM,LOC)                                   1
        IF(ITEM   .GT. 0 ) GO TO 602
 940    WRITE (MESG,9940)
 9940   FORMAT(5X,35H*** DOCUMENT FORMAT DOES NOT EXIST.)
        IERR=1
              GO TO 500
  602       IRDFI=IHDR(LOC)
C          READ THE DATA RECORD INTO  PRINT COMMON FOR USE IN'PRMON'ROUTINE
           I1=NCU* 48
           CALL DKGET(IRDFI,DFDRC,1,I1,NC          )
           GO TO 500
C


C
C          LINE NUMBERS
  100      IF(LNP .EQ. 1)GO TO 9111
           J=I+1
           IF ((PMPTR(J,3) .EQ. 5) .OR. (IHYPH .EQ. 1))GO TO 111
           GO TO 911
  111 IF (IHYPH) 110,11?,120
C-    HAVE A STARTING LINE NO.,PROCESS THE NO.
  110      IF ( LENG .LE. 8) GO TO  957
  112      WRITE( MESG,9955)                                                     1
 9955      FORMAT( 5X,40H*** LINE NUMBER MUST BE 1 TO  8 INTEGERS )              1
           IERR=1
          GO TO 500                                                              1
  957 ISLN = INTER(STRNG,IPTR,LENG)
      GO TO 500
C-    IF A HYPHEN ENCOUNTERED PREVIOUSLY WE HAVE ENDING LINE NO. (AREA FIELD).
  120  IF(LENG .LE. 8)GO TO 959
           WRITE(MESG,       9955)
          IERR=1
           GO TO 500
  959 IELN = INTER(STRNG,IPTR,LENG)
```

```
          IF(I .LT. ICNT)GO TO 500
C
C
CC
C
C
C-    CHECK TO SEE IF THE USER HAS GIVEN BOTH A START AND END LINE NUMBER.
C-    EITHER START OR END LINE HAS BEEN SPECIFIED. DETERMINE WHICH AND
C-   ISET DEFAULT FOR THE OTHER.
  510     JSLN=1
         JELN=99999999
            IF( ISEG(1) .EQ. ISEG(2) ) WRITE(MESG,5001 )
            IF( ISEG(1) .EQ. ISEG(3) ) WRITE(MESG,5001 )
            IF( ISEG(2) .EQ. ISEG(3) ) WRITE(MESG,5001 )
 5001       FORMAT ( 5X,32H***  DUPLICATE TEXT SEGMENT ID )
        IF (ISLN .LE. 0) ISLN = JSLN
        IF (IELN .LE. 0) IELN = JELN
          IF( ISLN . LT. JSLN ) ISLN= JSLN
          IF( IELN . GT. JELN ) IELN= JELN
C-    CHECK TO BE SURE THAT BOTH STARTING AND ENDING LINES EXIST.
  550     DO  557    J=1,4
          FILNM(J)= ITEXT(J)
  557     CONTINUE
        ISHID    = ITEXT(5)
        CALL TBLCS (ISHID, ISLN,IHDR,ITEM,LOC)
        IF(ITEM .GT. 0) GO TO 556
 930   WRITE(MESG,9930)
 9930  FORMAT(5X,40H*** STARTING LINE NUMBER DOES NOT EXIST.)
        WRITE(MESG,552) ISLN
  552  FORMAT(5X,16H*** LINE NUMBER   , I8 )
        ISLN = IHDR(-ITEM)
        WRITE(MESG,553) ISLN
  553  FORMAT(5X,46H*** DOES NOT EXIST AND HAS BEEN REPLACED WITH ,I8 )
  556    IRSLN=IHDR(LOC)
  551  CALL TBLCS (ISHID,IELN   ,IHDR,ITEM,LOC)
        IF(ITEM  .GT  0  ) GO TO 554
 931   WRITE (MESG,9931)
 9931  FORMAT(5X,38H*** ENDING LINE NUMBER DOES NOT EXIST.)
        WRITE(MESG,552) IELN
        IELN = IHDR(-ITEM)
        WRITE(MESG,553) IELN
  554  CONTINUE
         IRELN=IHDR(LOC)
C  CHECK FOR SLN GTR THAN ELN
        IF(ISLN  .LE. IELN ) GO TO 410
        WRITE(MESG,555)
  555  FORMAT( 5X,49H*** STARTING LINE NO. GREATER THAN ENDING NUMBER. )
         IERR=1
        GO TO 410
 9111        J=I+1
        IF(PMPTR(J,3) .EQ. 5) GO TO 9112
        IF(IHYPH .EQ. 1) GO TO 9112
        GO TO 911
&
C
C
C
C
```

```
C-     SPACING FIELD IS PRESENT.
  180 ISPAC = INTER(STRNG.IPTR.LENG)
  800  IF ((ISPAC  .GE.  1) .AND. (ISPAC  .LE. 2 ) )  GO TO 500                  1
        WRITE(MESG.9950)                                                        1
        ISPAC=2
  9950  FORMAT(5X.50H*** (INE SPACING INCORRECT. DOUBLE SPACING ASSUMED)        1
      GO TO 500
 9112 WRITE(MESG.9113)
 9113  FORMAT(5X.38H*** ONLY ONE LINE NUMBER PAIR ALLOWED        )
        IERR=1
           GO TO 500
C
C
C
C
  130  NITC = INTER(STRNG.IPTR.LENG)                                            1
       IF (( NITC  .GE.  0) .AND. ( NITC  .LE. 1 ) ) GO  TO 500                 1
        WRITE(MESG.9951)                                                        1
 9951   FORMAT(5X.42H*** INDEX CREATION INCORRECT. NONE CREATED)                1
           NITC=1                                                               1
       GO TO 500                                                                1
C
C
C
C
  140  LCO  = INTER(STRNG.IPTR.LENG)                                            1
       IF (( LCO   .GE.  0) .AND. ( LCO   .LE. 1 ) ) GO  TO 500                 1
        WRITE(MESG.9952)                                                        1
 9952   FORMAT(5X.46H*** LOGIC OVERRIDE INCORRECT. OVERRIDE ASSUMED)            1
          LCO=1                                                                 1
       GO TO 500                                                                1
C
C
C
C
  160  NOD  = INTER(STRNG.IPTR.LENG)
         IF( NOD  .EQ.  0 )   GO TO 500
         IF( NOD  .EQ.  1 )   GO TO 161
        WRITE(MESG.9953)                                                        1
 9953   FORMAT(5X.46H*** OUTPUT DEVICE INCORRECT. PRINTER ASSUMED)              1
  161    NPRI= NPRP
       GO TO 500                                                                1

C
C
C
  170  NPPIN= INTER(STRNG.IPTR.LENG)                                            1
       IF (( NPPIN .GE.  0) .AND. (NPPIN  .LE. 1 ) ) GO  TO 500                 1
        WRITE(MESG.9954)
 9954   FORMAT(5X.50H*** P/P INITIALIZATION INCORRECT. NONE PERFORMED  )        1
        NPPIN=1                                                                 1
       GO TO 500                                                                1
C
C
C
C
```

```
C
C-    ALPHA
   20  GO TO  (900.250.210.900.900.900.900.900).MODE
C
C
C
C
C
C-    OPTIONS FIELD ENCOUNTERED.
  250 CALL UNPAK(STPNG.IPTR.1.LETTR)
         IF (LENG   .LE. 1)  GO TO 961                                        1
         WRITE(MESG.9961)                                                     1
 9961    FORMAT (5X.43H*** COLUMN CODE GREATER THAN ONE CHARACTER.  /          1
     1                5X.34H*** ONLY THE FIRST CHARACTER USED.          )      1
  961    CONTINUE                                                             1
      DO 260 J=1.6
      IF (LETTR-FLGNM(J)) 260.270.260
  260 CONTINUE
        GO TO 900
C-    RECORD OPTION ENTERED.
  270 FLGOP(J)=1
      IF (IHYPH) 500.500.275
  275 FLGOP(J)=-1
      IHYPH=0
      GO TO 500
C
C

C-    AREA FIELD - TEXT SEGMENT IS PRESENT.
  210 J=I+1
       IF(PMPTR(J.3) .EQ. 5) GO TO 112
        IF(IHYPH .EQ. 1) GO TO 112
      IF(LENG-4   ) 220.220.911
  911  WRITE (MESG.9911)                                                      1
 9911  FORMAT ( 5X.47H*** TEXT SEGMENT ID MUST BE 1-4 A/N CHARACTERS. )        1
         IERR=1
      GOTO 500
  220  IS= IS+1                                                               1
       IF(IS .LE. 3 )  GO TO 221                                              1
       WRITE(MESG.9912)                                                       1
 9912  FORMAT(5X.41H*** ONLY THREE TEXT SEGMENT IDS PERMITED.    )             1
       IERR= 1                                                                1
       GO TO 500                                                              1
  221  CALL ICOPY (STRNG.IPTR.LENG.ISEG(IS).1)                                1
      GO TO 500
C
C
C
C


C
C-    SEMICOLON ENCOUNTERED.
   40  GO TO (410.410.510.410.410.410.410.920).MODE
C-    PROCESSING OF THIS PARAMETER COMPLETE.  MOVE TO NEXT PARM. RESET KEYS
  410 MODE=MODE+1
      IMERR=0
      IHYPH=0
      GO TO 500
C
```

```
C
C
C
C
C-     HYPHEN ENCOUNTERED
   50   GO TO (900,450,450,900,900,900,900,900),MODE                           1
  450  IHYPH=1
       GO TO 500
C
C
C
C
C
C-     COMMA ENCOUNTERED
   60   GO TO (900,500,610,900,900,900,900,900),MODE
  610   LNP=1
         IHYPH=0
C
C
C
C
C
C-     END OF THE PARAMETER PROCESSING
  500  CONTINUE
C
C
C
C
C
C
C
         IF(IFORM .EQ. 0) GO TO 991
C-     IF TEXT ONLY IS TO BE PRINTED, SET UNSELECTED COLUMNS TO NOT PRINT.
C-     SET DEFAULT PRINT OPTIONS
  700  ISET=1
       IF (FLGOP(1)) 720,720,710
  710  ISET=-1
  720  DO 730 J=2,6
       IF (FLGOP(J)) 730,725,730
  725  FLGOP(J)=ISET
  730  CONTINUE
       FLGOP(1)=1
C-     DEBUG OPTION PRINT COMMANDS.
        IF (DEBUG-2) 9997,9998,9998
 9998  WRITE(MESG,9999) ISLN,IELN,IFORM,FLGOP,ISPAC,IERR                DEBUG
9999   FORMAT(10I10)                                                    DEBUG
C-     CHECK TO SEE IF ERRORS FOUND.
 9997  IF(IERR) 850,850,990
850    CONTINUE
       CALL PRMON(ISLN,IELN,IFORM,FLGOP,ISPAC,           IRDFI,IRSLN,
     1 IRELN)
1000   CONTINUE
        DO 603 J=1,4
 603         FILNM(J)=EDITF(J)
       RETURN
C
C
```

```
C
C
C
C
C
C     PRINT FORMAT ERROR ONCE PER FIELD
900   IF(IMERR) 901,901,960
901   IMERR=1
      IERR=1
      WRITE (MESG,9901)
9901  FORMAT(5X,20H*** FORMAT ERROR IN  )
       GO TO ( 903 ,904, 902, 905 , 906 ,907 , 908,909),MODE
902   WRITE(MESG,9902)
9902  FORMAT(25X,80H AREA FIELD.  MUST BE SLN-ELN(INTEGERS) AND/OR 3 TEX
     1T SEGMENTS (ALPHANIMERIC).                            )
       GO TO 960                                                        1
903   WRITE(MESG,9903)
9903  FORMAT(25X,60H DOCUMENTM FORMAT FIELD.  MUST BE A 1 TO 8 CHARACTER
     1 INTEGER.       )
       GO TO 960                                                        1
904   WRITE(MESG,9904)
9904  FORMAT(25X, 52H COLUMNS TO BE PRINTED. MUST BE F,L,T,A,X,P,OR - .
     1        )
       GO TO 960                                                        1
905   WRITE(MESG,9905)
9905  FORMAT(25X,40H LINE SPACING FIELD.  MUST BE 1 OR 2.        )
       GO TO 960                                                        1
  906  WRITE(MESG,9906)                                                 1
 9906  FORMAT(25X,55H NO INDEX AND TABLE CREATION FIELD MUST BE 0 OR 1.
     1       )
       GO TO 960                                                        1
  907  WRITE(MESG,9907)                                                 1
 9907  FORMAT(25X,52H LOGIC CONDITION OVERRIDE FIELD. MUST BE 0 OR 1.
     1           )
       GO TO 960                                                        1
  908  WRITE(MESG,9908)                                                 1
 9908  FORMAT(25X,48H OUTPUT DEVICE NUMBER FIELD. MUST BE 0 OR 1.      )
       GO TO 960                                                        1
  909  WRITE(MESG,9909)                                                 1
 9909  FORMAT(25X,67H NO PAGE/PARAGRAPH NUMBER INITIALIZATION FIELD. MUS
     1T BE 0 OR 1.                       )
       GO TO 960                                                        1
C
C
C
```

NOTE USE OF STANDAR
FORMAT ERRORS FOR
1. FIELD
Then
1.1 TYPE encountered.

FIELDS

```
c
 960  GO TO ( 9001,9002,9003,9004,9005,9006,9007,9008  ),ITYPE
C
C
C
C
 9001  WRITE(MESG,99001)
99001  FORMAT (5X,22H*** NUMERIC NOT VALID.   )
       GO TO 500
 9002  WRITE(MESG,99002)
99002  FORMAT (5X,27H*** ALPHANUMERIC NOT VALID.   )
       GO TO 500
 9003  WRITE(MESG,99003)
99003  FORMAT (5X,37H*** ALPHANUMERIC DELIMITED NOT VALID.   )
       GO TO 500
 9004  WRITE(MESG,99004)
99004  FORMAT (5X,25H*** SEMI-COLON NOT VALID.  )
       GO TO 500
 9005  WRITE(MESG,99005)
99005  FORMAT (5X,20H*** HYPHEN NOT VALID        )
       GO TO 500
 9006  WRITE(MESG,99006)
99006  FORMAT (5X,20H*** COMMA  NOT VALID        )
       GO TO 500
 9007  WRITE(MESG,99007)
99007  FORMAT (5X,28H*** DOUBLE ASTERIX NOT VALID    )
       GO TO 500
 9008  WRITE(MESG,99008)
99008  FORMAT (5X,28H*** PLUSS SIGN NOT VALID.                )
       GO TO 500
C
C
C
C
C
C
C
920   WRITE(MESG,9920)
9920  FORMAT(5X,26H*** TOO MANY FIELDS GIVEN.)
        IF(IFORM .EQ. 0) GO TO 991
      IERR=1
        GO TO 990
 991   WRITE (MESG,9991)
 9991  FORMAT(5X, 50H*** NO FIELDS FOUND.  DOCUMENT FORMAT ID REQUIRED.)
       GO TO 990
 992   WRITE (MESG,9992)
 9992  FORMAT(5X, 50H*** TOO MANY FIELDS FOUND.  PROCESSING CONTINUES. )
       GO TO 5
990   WRITE(MESG,9990)
9990  FORMAT(5X,46H*** COMMAND IGNORED, BUT PROCESSING CONTINUES.  )
      GOTO 1000
      END
```

1-8

Note transfer to end of Do Loop for parameter processing.

APPENDIX D

PROGRAMMING NOTES & PROCEDURES

D.1 WORK PROGRESSION

1. Users Manual Description

a. Review Functional Criteria.

b. Discuss Functional Criteria with Supervisor.

c. Check all other commands for possible interactions.

d. Review table structure and command with supervisor.

e. Review code for additional knowledge.

f. Design and write the final description

g. Supervisor review of description.

h. Typing.

i. TIB review.

g. Drafting.

2. Complete Coding

a. Review existing code.

b. Document code to standards.

c. Design coding changes to complete the command. Update table descriptions.

d. Code the changes.

e. Design a 100% complete test deck that will operate as a stand-alone run.

f. Hand debug the test run thoroughly.

g. Review the test run and hand calculations with your supervisor.

h. Run on the computer.

3. Reporting Standards

a. Each team member will keep a daily log book. This log will detail the time spent on the project. The log book will be daily dated and entries made recording start and completion times. Computer run costs will be entered directly into the log. Log books will be checked by the supervisor no later than COB each Friday.

b. All computer trailer sheets must be saved and numbered by the programmer. Supervisor's permission should be granted before submission of a computer run.

c. All decisions must be recorded in writing within the log books of all parties involved in the system.

d. All considerations reviewed must be entered into the log book for future reference. This log will become a detailed daily diary of your work.

D.2 DOCUMENTATION STANDARDS

These standards apply to all programming performed.

All program documentation should be contained in comments in the programs. The comments which will be included in the documentation should be in the format described below.

The resulting listed documentation will be in the following format:

Program Name: as in the first line of the function or subroutine, with arguments.

Function: brief description of the function of the program.

Author: author's name, date of first writing.

Modifications: Each mod should be numbered. Author, date and brief description of the reason for the change(s). All cards associated with each mod should be so numbered in c.c. 79-80.

Language: programming language.

Calling Sequence: description of arguments and function result.

Routines Called: list of subroutines and functions called by the program.

Tasks or Modules: tasks or modules containing this routine.

Variables: a description of all variables used in the program. COMMON variables will generally be described in a "main" program, referenced in this section. Check to insure that the command description is correct. The structure of arrays and meaning of variables should be fully described.

Any figures or drawings which would aid in the description of an algorithm should be named with the program name and a figure number, such as ABC.1 for program ABC. These are referenced from the comments in the subroutine and included in the documentation book.

D.3 TECHNICAL DOCUMENTATION GUIDELINES

A. Contents

1. All textual documentation will be contained in comments in the programs. It will be possible to obtain a listing of the appropriate comments separately.

B. FORM

1. Placement of comments
   All comments start in cc7.

2. Separating sections
   Blank comment lines and lines of asterisks may be used to separate sections of code.

   a. DO loops - A blank comment line before the DO statement and a blank comment line after the CONTINUE statement is used to 'bracket' major DO loops. Comments pertaining to the loop are included within the 'bracket'.

   b. Major headings are 'bracketed' by a blank comment line before and after the headings.

   c. A line of asterisks may be used to signify the end of a major program section.

   d. Headings may be caused to stand out further by inserting a space after each character and by using a line of asterisks right, left, above, or below the heading.

Note: All the above devices are used to enhance clarity for anyone reading the program. Uniform use of format within each program system will amplify the effectiveness of same.

3. Statement numbers.

a. A FORMAT statement follows the first WRITE or READ statement using it.

b. Error messages start at statements numbered 99, or 999 or 9999.

c. RETURN is statement number 100, 1000, or 10000.

d. Statement numbers can be sequenced in two ways: 1) Increasing value sequence, 2) Collating sequence. If increasing sequence is used, the statement numbers should be right justified. Some people reserve 50's for ending DO loops, and derive the number of a FORMAT statement from the associated READ or WRITE statement by adding 'one'. <u>Read</u> or <u>Write</u> statements are frequently numbered in hundreds:

```
1800 WRITE(3,1801)x,y,z
1801 FORMAT(3F8.4)
```

In a long program with many READs and WRITEs 1802,3, etc. would be preferable to using 1821, 1822, etc., (If one is in danger of using up all available hundreds, that is.) If Collating sequence is used, the statement numbers should be left justified. 50's might be reserved for ending DO loops. FORMAT numbers may be derived from associated READ or WRITE statement numbers of multiplying with 10.

4. Debugging statements.
For automatic removal of tracers, traps, etc., an asterisk in cc2 followed by the number of lines to be removed from the final program might be used.

C. FORTRAN Standards

The American National Standard FORTRAN (ANSI x3.9-1966) standards are followed throughout.

D. Additional Guidelines

1. **FUNCTIONS may not define or redefine any arguments or variables in COMMON.**

2. **DIMENSION and TYPE statements come before COMMON, which come before EQUIVALENCE statements.**

3. **No recursive procedures will be written.**

4. **All DO loops end on unique CONTINUE statements.**

5. I/O device unit members should be variables in COMMON.

6. All specifications statements appear before all executable statements.

7. Each subroutine or function subprogram has a single entry point, at the beginning, and a single RETURN statement as the last executable statement.

8. Avoid word-size-, character-size-, or absolute-address-dependent operations. Similarly, do not assume the format of negative integers (1's or 2's complement) or the internal representation of character or numeric data.

9. Profuse comments should be used.

E. Programming "Tricks"

1. Keep all statement numbers in order. This can be done by considering all statement numbers having fewer than 5 digits as being filled with trailing zeroes. This will guarantee that statement numbers are not duplicated and that a programmer does not have to search throughout a program to find if a particular statement number has been used.

2. Filler variables in COMMON should be named according to their word location in COMMON. This allows for easy computation of the size of filler arrays and insertion of new variables in COMMON.

3. Subroutines should be kept to less than 100 executable statements if possible. Individual subroutines should each serve only a single function.

4. Error and warning messages should state in the user's terms what the problem is and what action is taken by the program.

5. Intermediate output for additional data should be available by changing an input variable. This variable will set dump switches which are either stored or are accessible through COMMON.

F. Program Conventions

1. If a transfer within an IF statement is impossible under normal data transfer to an error message that states

   "The computer program has transferred to an impossible location in routine number _____"

2. An error message, written in English phrases, should give the user the same information that the computer has. The object is to provide maximum service to the user and to eliminate all confusion.

3. During debugging list the subroutine parameters (1) before the call, (2) immediately inside the subroutine, (3) before returning, (4) immediately after the return from the call in culling program. Place IF statements around the debugs and mark in Col 73-80 as GUBED-C.

4. Do not remove existing coding cards from the deck. Mark a C in column one and retain.

5. Mark your card entries with the number of the modification in card columns 77-80.

6. Standard error message prefix will be (5x,--Hxxx.

7. All common blocks will be set up as follows:

a. Date Modified

b. Common Statements

c. Dimension Statements

d. Equivalence Statements

e. Integer Statements

f. Comment cards defining each variable in order

8. Array ITNM (3,8) has been added to /SYSTM/ common for writing error messages. The columns correspond to the parameter field type values in PMPRTR (,3).

9. All system routines should set FILNM to SYSFL before returning. All edit routines should set FILNM to EDITF before returning.

APPENDIX E

PROGRAM COMMON AREAS

```
C
C         JULY 22, 1977
        INTEGER FILNM(4)
        INTEGER RVRSE,TRU,FALS,ZEROS,ONES,FOLID,PREID,FSTAT,P
        COMMON P(200)
        EQUIVALENCE (P(5),NBW),(P(6),NCW),(P(7),NAW),(P(8),NBC),(P(9),NCU)
        EQUIVALENCE (P(10),NBU),(P(11),NAU),(P(12),NBU2),(P(13),NC2U)
        EQUIVALENCE (P(14),RVRSE),(P(15),NWPSU),(P(16),LFIW),(P(17),LFIWU)
        EQUIVALENCE (P(24),ICRD),(P(25),IPRT),(P(26),IPCH),(P(27),ITAP)
        EQUIVALENCE (P(38),TRU),(P(39),FALS),(P(40),ZERCS),(P(41),ONES)
        EQUIVALENCE (P(187),MAXGP),(P(188),NDS),(P(189),FILNM(1))
        EQUIVALENCE (P(193),FOLID),(P(194),PREIC),(P(195),LRLEN)
        EQUIVALENCE (P(196),FSTAT),(P(197),NDA),(P(198),IDHTR)
        EQUIVALENCE (P(199),IDHER),(P(200),IDKER)
C
C       NBW IS NUMBER OF BITS IN A MACHINE WORD (A NON-FORTRAN CONCEPT)
C                 A MACHINE WORD IS USUALLY EQUAL TO A STANDARD-UNIT.
C                 A STANDARD UNIT IS THE AMOUNT OF STORAGE NEEDED FOR AN INTEGER
C       NCW IS NUMBER OF CHARACTERS IN A MACHINE WORD
C       NAW IS NUMBER OF MACHINE ADDRESSES THAT WILL FIT IN A MACHINE WORD
C                 THIS IS PROBABLY NEVER USED
C       NBC IS NUMBER OF BITS PER CHARACTER.
C       NCU IS NUMBER OF CHARACTERS PER STANDARD-UNIT (ATLEAST 4 FOR EDITSPEC)
C       NBU IS NUMBER OF BITS PER SU (STANDARD-UNIT)
C       NAU IS NUMBER OF ADDRESSES PER SU. PROBABLY UNUSED.
C       NBU2 IS NBU/2
C       NC2U IS NCU*2
C       RVRSE DENOTES ORDER OF CHARACTERS IN AN INTEGER VARIABLE.
C                 0 = LEFT TO RIGHT (A1 REFERS TO HIGH-ORDER OR LEFTMOST BITS)
C                 1 = RIGHT TO LEFT (A1 REFERS TO LOW-ORDER OR RIGHTMOST BITS)
C       NWPSU IS NUMBER OF WORDS PER SU (USUALLY 1)
C       LFIW  IS NUMBER OF BITS IN A FORTRAN INTEGER (SAME AS NBU)
C       LFIWU IS NUMBER OF USABLE BITS IN A FORTRAN INTEGER.
C                 MOST OF THE ABOVE VARIABLES EXIST FOR HISTORICAL REASONS
C                 ONLY. PROGRAMMERS SHOULD NOT NEED ANYTHING BUT NCU OUT OF THESE.
C       ICRD IS CARD-READER LUN (PROBABLY NOT USED)
C       IPRT IS PRINTER LUN (PROBABLY NOT USED)
C       IPCH IS PUNCH LUN (PROBABLY NOT USED)
C       ITAP IS MAGNETIC TAPE LUN (PROBABLY NOT USED)
C                 THE ABOVE LOGICAL-UNIT-NUMBERS EXIST FOR HISTORICAL REASONS
C                 FOR THE LUN VARIABLES USED IN EDITSPEC. SEE COMMON /IOC/
C       TRU IS AN INTEGER REPRESENTATION OF LOGICAL .TRUE.
C       FALS IS AN INTEGER REPRESENTATION OF LOGICAL .FALSE.
C       ZEROS IS AN INTEGER REPRESENTATION OF BINARY ZERO (ALL BITS OFF)
C       ONES IS AN INTEGER REPRESENTATION OF BINARY ONES (ALL BITS ON).
C                 THE ABOVE WERE USED IN THE OLD SYSTEM TO PERFORM SOME
C                 TRICKY TYPE CONVERSION. THEY CONTINUE TO EXIST SOLELY
C                 FOR HISTORICAL REASONS.
C       MAXGP STANDS FOR MAXIMUM NUMBER OF CHARACTERS PER GET OR PUT. IT
C                 WAS USED IN THE EARLIER DATA-HANDLER FOR I/O. IT IS
C                 CURRENTLY SUPERFLUOUS, AND HAS A VALUE OF 32768 (2**15)
C
C                 THE FOLLOWING ARE USED TO COMMUNICATE BETWEEN EDITSPEC
C                 AND THE DATA-HANDLER.
C
C       NDS   - DATASET NUMBER (INDEX INTO DATASET TABLE)
C       FILNM - 4 INTEGER ARRAY TO STORE FILE-NAME
C       FOLID - RECORD ID OF SUBSEQUENT OR FOLLOWING RECORD
C       PREID - RECORD ID OF PREVIOUS RECORD
C       LRLEN - LOGICAL RECORD LENGTH
C       FSTAT - FILE STATUS
C       NDA   - MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
```

```
C       NDA    - MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C       IDHTR - DATA-HANDLER TRACE FLAG
C       IDHER - DATA-HANDLER ERROR FLAG
C       IDKER - DATA-HANDLER ERROR CODE
C
C


C
C
C       DECEMBER 1, 1977
C       ACCESS SYNTAX COMMON.
        INTEGER DNAMS,ACODE,ANUMS,UIDS,UALLS,ALLSW,RCDID,RCAID,RCUID
        COMMON /ACSNC/ DNAMS(3,20),ACODE(20),ANUMS(3,20),UIDS(3,20),
       1UALLS(3,20),NNAMS,NNUMS,NIDS,NALLS,ALLSW,RCDID(20),RCAID(20),
       2RCUID(20)
C
C                   DNAMS(3,20) IS USED TO STORE UPTO 20 DOCUMENT NAMES.
C                   ACODE(20) IS ACCESS CODE FOR DOCUMENT NAME LIST.
C                   ANUMS(3,20) IS LIST CF ACCOUNT NUMBERS.
C                   UIDS(3,20) IS LIST OF USER-IDS.
C                   UALLS(3,20) IS LIST CF ALL-MASKS.
C                   NNAMS IS NUMBER OF DOCUMENT NAMES IN LIST (<=20)
C                   NNUMS IS NUMBER OF ACCOUNT NUMBERS.
C                   NIDS IS NUMBER OF USER-IDS.
C                   NALLS IS NUMBER OF ALL MASKS.
C                   ALLSW IS 1 IF 'ALL' IS SPECIFIED.
C                   RCDID(20) HOLDS RECORD-ID OF FIRST DATA RECORD ASSOCIATED
C                             WITH EACH DOCUMENT NAME IN DNAM(3,20) (IN DIR-TABLE).
C                   RCAID(20) IS DATA RECORD ASSOCIATED WITH ANUMS(3,20) IN ACS-TBL
C                   RCUID(20) IS DATA RECORD ASSOCIATED WITH UIDS(3,20) IN USR-TBL.
C
C

C AUGUST 30,1979
          COMMON /ACTAC/ IACTA
C IACTA =0 OFF,=1 WHEN A *CT* FOUND INSIDE A *TB*

C         MARCH 8, 1979
C       BACKUP COMMON BLOCK
        INTEGER STAPE
        COMMON /BACKC/ ISBUP(5), ISBUC(5), ISBUT(5), MULFIL,
       1                NUNITS, STAPE,
       1IRSTR,IDOCDS
C       ISBUP IS TABLE OF ENTITIES EDITED SINCE LAST BACKUP
C       ISBUC IS BACKUP DIRECTORY BY DOCUMENT/DATASET/SYSTEM-TABLE NAME
C       ISBUT IS BACKUP DIRECTORY BY TAPE NAME
C       MULFIL IS TYPE OF BACKUP TAPE USED
C          =0 IF THERE IS ONLY ONE FILE ON THE TAPE
C          =1 IF THERE MAY BE MORE THAN ONE FILE ON THE TAPE
C       NUNITS IS TOTAL NUMBER OF THINGS THAT CAN BE BACKED UP
C          =MAXIMUM NUMBER OF FILES ON THE TAPE IF MULFIL=1
C          =MAXIMUM NUMBER OF ENTITIES THAT MAY BE BACKED UP IF MULFIL=0.
C       STAPE IS UNIT NUMBER OF THE "SPECIAL TAPE" TO HOLD ISBUP,ISBUC, AND ISBUT.
C       IRSTR - RESTORE SWITCH

C
C         APRIL 12, 1979
C         COMMON HOLDS NDS AND FILNM FOR RESTORE READ ROUTINE
C
        INTEGER RENDS, REFIL
       1, RRECID
       2, INISAV
C          RRECID RECORD ID TO READ NEXT
C          INISAV   SAVE AREA FOR INPI
        COMMON /BCKRSC/ RENDS, REFIL(4)
       1, RRECID
```

```
C
C
C
C
C
C          CONVERSION-SYSTEM COMMON, DOCUMENT PARTITION
C
C          FEBRUARY 6, 1979
C
           COMMON /CNVDOC/ DOCNM, DSNAM, NEWDSN, OPTION, CMDOPT,
                           PROCES, CNVERR, DSCAT, DSKEY, DSRID,
                           DSBUF, SEQNO
C
                 INTEGER   DOCNM(3)
C                                    (PACKED) EDITSPEC NAME OF DOC TO
C                                     BE CONVERTED
C
                 INTEGER   DSNAM(2)
C                                    DATASET-NAME ON WHICH THE DOCUMENT
C                                     RESIDES.
C
                 INTEGER   NEWDSN(2)
C                                    TARGET DATASET-NAME FOR THIS
C                                     DOCUMENT. IF USER CHOOSES TO
C                                     MODIFY THE DATASET THAT THE
C                                     DOCUMENT RESIDES ON AT TAPE
C                                     GENERATION TIME
C
                 INTEGER   OPTION
C                                    SWITCH THAT CONTAINS THE OPTION
C                                     VALUE THE USER SELECTED WHEN
C                                     THE ".DSCN" COMMAND WAS ISSUED.
C
                 INTEGER   CMDOPT
C                                    OPTION SELECTED ON THE SPECIAL
C                                     COMMAND RECORDS THAT FOLLOW THE
C                                     ".DSCN ... ." COMMAND WITH OPT
C                                     SET TO 2.
C
                 LOGICAL   PROCES
C                                    LOGICAL SWITCH TO INDICATE WHETHER
C                                     THE "LOOKUP" PHASE HAS SELECTED
C                                     ANOTHER DOCUMENT TO BE MOVED TO
C                                     THE CONVERSION MEDIUM.
C                                     (IF FALSE, INDICATES TO TERMINATE
C                                     CONVERSION PROCESSING).
C
                 LOGICAL   CNVERR
C                                    SWITCH TO INDICATE THAT THE CONV
C                                     PROCESS HAS FAILED
C
                 INTEGER   DSCAT(5)
C                                    TABLE IDENTIFIER TO ACCESS THE
C                                     SUB-TABLE THAT INDICATES THE
C                                     DOCUMENT CATALOG FOR A PARTICULAR
C                                     DATASET.
C
                 INTEGER   DSKEY
C                                    SUBSCRIPT IN DSBUF() WHERE THE
C                                     FIRST WORD OF THE SELECTED KEY
C                                     (DOCUMENT NAME) IS TO BE FOUND
C
                 INTEGER   DSRID
C                                    SUBSCRIPT IF DSBUF() WHERE THE RECORD-ID
C                                     OF THE RECORD ASSOCIATED WITH
```

```
C      NDA   - MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C                                         OF THE RECORD ASSOCIATED WITH
C                                         THE KEY @ DSKEY IF FOUND
C
                 INTEGER   DSBUF(243)
C                                        BUFFER AREA FOR HEADER RECORDS IF
C                                         THE ISDDT TABLE ACCESSES WHILE
C                                         DOCUMENT TRANSFERS ARE BEING
C                                         PERFORMED
C
                 INTEGER   SEQNO
C                                        SEQUENCE NUMBER OF THE NEXT CONV
C                                         RECORD TO BE GENERATED
C
C

C
C
C         CONVERSION SYSTEM GENERAL COMMON
C
C         AUGUST 3, 1979
C
          COMMON /CONVRC/ ITONR, CVLUN, ISDDT, CVDLIM, CVBUF, CVCNT
C
               INTEGER   ITONR(5)
C                                  TABLE IDENTIFIER FOR RECORD-ID TRANSATION
C                                   TABLE.
C
               INTEGER   CVLUN
C                                  LOGICAL UNIT NUMBER FOR CONVERTION DATA
C                                   OUTPUT.
C
               INTEGER   ISDDT(5)
C                                  TABLE IDENTIFIER FOR THE DATASET-NAME
C                                   TABLE.
C
               INTEGER   CVDLIM
C                                  MACHINE (A1) REPRESENTATION OF THE
C                                   CHARACTER TO BE USED AS THE "SPECIAL
C                                   CHARACTER DELIMITER" FOR ALL EDITSPEC
C                                   COMMANDS ISSUED BY THE CONVERTION SYSTEM.
C
               INTEGER   CVBUF(400)
C                                  GENERAL BUFFER AREA FOR USE BY THE CONVERTION
C                                   SYSTEM.
C
               INTEGER   CVCNT
C                                  COUNT OF ALL CONVERTION RECORDS
C                                   GENERATED BY THE CONV SYSTEM
C
C
C


            COMMON /CONVR2/
          1ZLOGON(     6),ZDOCU (   54),ZFOOT (  102),IDUMMY(  245),
          1JDUMMY(   245),ISDATA(   40),IUDATA(   30),ISPECP(  625),
          1IKEY  (     3)
            INTEGER ZDOCU,ZFOOT
C          AUGUST 31, 1977
           INTEGER INPCS,OUTCS
           COMMON /CSETC/ INPCS,OUTCS
C
C          INPCS IS INPUT CHARACTER SET.
C          OUTCS IS OUTPUT CHARACTER SET.
C
C
```

```
C
C       OCTOBER 27, 1978
C
C       CONVERTION-SYSTEM LOGIN COMMON
C
        COMMON /CVLGNC/ LOGIDE
C
             INTEGER   LOGIDE(3)
C                                   CONTAINS THE USER-ID (IN EDITSPEC FORM)
C                                    THAT WAS USED IN THE LAST CALL TO
C                                    "CVLGIN".   SHOULD BE DATAED IN BLOCK DATA
C                                    TO / 3*0 /
C
C


C
C
C       COMMON BLOCK:  CVUSRC          06-SEP-78
C
C
        COMMON /CVUSRC/     SEQNM,  RECSIZ, DATALN
C
C
C       COMMON VARIABLE DESCRIPTIONS:
C
C            TYPE      NAME      DESCRIPTION
C
C

        INTEGER   SEQNM
C                                   CONTAINS THE SEQUENCE NUMBER OF
C                                    THE NEXT CONVERSION RECORD TO
C                                    BE EMITTED.
C
        INTEGER   RECSIZ
C                                   CONTAINS THE RECORD SIZE (IN
C                                    CHARACTERS) THAT WAS READ
C                                    DURING THE LAST DKGET.  USED
C                                    PRIMARILY BY CVUSR2.
C
        INTEGER   DATALN(68)
C                                    BUFFER TO HOLD THE ALPHAMERIC
C                                    PORTIONS OF THE CONVERSION
C                                    RECORDS.
C
C


C
C
C      JULY 22, 1977
       INTEGER DEBUG
       COMMON /DEBUGC/DEBUG
C    DEBUG IS TRACE OUUPUT SWITCH.
C
C
C
C
C    DOCUMENT KEYWORD INDEX TABLE COMMON
C    MAY 25, 1978
     INTEGER DKYWDT
     COMMON /DKYWDC/ DKYWDT(5,6),KWUSE
C    DKYWDT: DOCUMENT KEYWORD TABLE
C   DKYWDT(1,N)= THE PARTICULAR PART OF THE FILE-NAME
C   DKYWDT(2,N)= THE ID  OF THE FIRST RECORD
C   DKYWDT(3,N)= THE SIZE OF THE PRIMARY KEY
C   DKYWDT(4,N)= THE SIZE OF THE SECONDARY KEY
```

```
C       NDA   - MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C      [illegible] THE SIZE OF THE SECONDARY [illegible] ASSOCIATED WITH
C      DKYWDT(5.N)= THE STORAGE MODE (PACK/POINT)
C       WHERE N IS THE KEYWORD INDEX (1 TO 6)
C       KWUSE=NUMBER OF KEYWORD INDEX TABLES USED BY THE DOCUMENT
C
C
C


C
C         JULY 22. 1977
C       DOCUMENT EDIT COMMON
        INTEGER EDITF.EDS.EBDS.EACC.EREL
        LOGICAL EREAD.EDBAK
        COMMON /EDITC/ EDITF(4).EDS.EBDS.EREAD.IEDIT.MGO.ICYC.IDAC.INC.
       1                  ITOTY(5.15).IERR.IAUD.EDBAK.EACC.EREL
        DIMENSION ITAUG(5).ITAUO(5).ITAUT(5).ITDRT(5).ITERT(5).
       1          ITERC(5).ITERL(5).ITERS(5).ITFLG(5).ITIX (5).
       2          ITLC (5).ITPUL(5).ITABC(5).ITEXT(5).ITBAK(5)
        EQUIVALENCE (ITAUG(1).ITOTY(1. 1)).(ITAUO(1).ITOTY(1. 2))
        EQUIVALENCE (ITAUT(1).ITOTY(1. 3)).(ITDRT(1).ITCTY(1. 4))
        EQUIVALENCE (ITERT(1).ITOTY(1. 5)).(ITERC(1).ITOTY(1. 6))
        EQUIVALENCE (ITERL(1).ITOTY(1. 7)).(ITERS(1).ITCTY(1. 8))
        EQUIVALENCE (ITFLG(1).ITGTY(1. 9)).(ITIX (1).ITCTY(1.10))
        EQUIVALENCE (ITLC (1).ITOTY(1.11)).(ITPUL(1).ITCTY(1.12))
        EQUIVALENCE (ITABC(1).ITOTY(1.13)).(ITEXT.1).ITOTY(1.14))
        EQUIVALENCE (ITBAK(1).ITOTY(1.15))
C       EDITF IS AN ARRAY CONTAINING THE DOCUMENT-NAME OF THE DOCUMENT
C                BEING EDITED. IN THE FIRST 3 WORDS. AND ALL BLANKS IN
C                THE 4TH WORD
C       EDS IS THE DATASET NUMBER OF THE DATASET CONTAINING THE DOCUMENT
C                BEING EDITED
C       EBDS IS THE DATASET NUMBER OF THE DOCUMENT COMMAND BACKUP DATASET
C       EREAD IS REA-ONLY-FLAG.
C                .FALSE. MEANS READ/WRITE ACCESS TO DOCUMENT
C                .TRUE. MEANS READ-ONLY ACCESS.
C       IEDIT IS IN-EDIT-MODE SWITCH.
C                0 MEANS NOT-IN-EDIT-MODE
C                1 MEANS IN-EDIT-MODE
C       MGO IS XTRNL-REFERENCE-OVERRIDE SWITCH
C                0 MEANS DO NOT OVERRIDE
C               -1 MEANS OVERRIDE
C       ICYC IS CURRENT CYCLE NUMBER
C       IDAC IS USER DOCUMENT ACCESS CODE
C       INC IS LINE INCREMENT
C       ITOTY IS 15 DIFFERENT ARRAYS. EACH 5 WORDS LONG - ONE FOR EACH OF
C                THE 14 DIFFERENT DOCUMENT TABLES. AND ONE FOR THE BACKUP
C                TABLE. FOR EACH ARRAY. THE 5 WORDS HAVE THE FOLLOWING
C                INFORMATION-
C                1. THE PARTICULAR PART OF THE FILE-NAME (FILNM(4))
C                2. THE ID OF THE FIRST RECORD
C                3. SIZE OF PRIMARY KEY
C                4. SIZE OF SECONDARY KEY
C                5. STORAGE MODE (PACK OR/AND POINT)
C       THE 15 DIFFERENT ARRAYS ARE AS FOLLOWS. FOR EACH TABLE. THE FORTRAN
C                VARIABLE NAME. THE PARTICULAR PART OF THE FILE-NAME OF
C                THE TABLE. AND A DESCRIPTION OF THE TABLE.
C
C
C       IT---     VALUE OF IT---(1)        DESCRIPTION
C                    (FILNM(4))
C       ------------------------------------------------------------
C
C
C
C       ITAUG          'AUG'               AUDIT TABLE (GENERAL)
C       ITAUO          'AUO'               AUDIT TABLE (OTHER)
C       ITAUT          'AUT'               AUDIT TABLE (TEXT)
C       ITDRT          'DRT'               DOCUMENT REFERENCED TABLE
```

```
C      ITERT             .ERT.                    EXTERNAL REFERENCE (TABLE)
C      ITERC             .ERC.                    EXTERNAL REFERENCE (COPY)
C      ITIX              .IX.                     INDEX
C      ITLC              .LC.                     LOGIC CHECK
C      ITPUL             .PUL.                    PULL TABLE
C      ITABC             .ABC.                    TABLE OF CONTENTS
C      ITEXT             .EXT.                    TEXT TABLE
C      ITBAK             .BAK.                    BACKUP TABLE
C      EACC=1 FOR READ-WRITE; 2 FOR READ-PRINT; 3 FOR READ-ONLY.
C      EREL IS 1 IF LOCKED EDIT DOCUMENT CAN BE RELEASED BETWEEN COMMANDS.
C


C
C
C      AUGUST 31. 1977
       LOGICAL EDEL.EOLIN
       INTEGER EXDOC.EXDS.ETEXT.EHDR.EFLAG.ELOC.ELNST.ELNNO.ELNND.EXECSW
       COMMON /EXECC/ EXDOC(3).EXDS.ETEXT(5).EHDR(243).EFLAG.ELOC.
      1ELNST.ELNNO.ELNND.EXECSW.EDEL.EOLIN
C
C      EXDOC IS DOCUMENT FROM WHICH COMMANDS ARE EXECUTED FOR .EXEC. COMMAND.
C      EXDS IS DATASET NUMRER OF EXDOC.
C      ETEXT IS 5 WORD TABLE PARAMETEP ARRAY, SIMILAR TO ITEXT(5)
C      EHDR(243). EFLAG. ELOC ARE FOR TABLE-HANDLING ROUTINES.
C      ELNST IS STARTING LINE NUMBER.
C      ELNNO IS CURRENT LINE NUMBER. FOR NEXT COMMAND
C      ELNND IS ENDING LINE NUMBER.
C      EXECSW IS SWITCH:   1 FOR NORMAL READ
C                          2 FOR .EX. COMMAND
C                          3 FOR .EXEC. COMMAND.
C
C


C     OCTOBER 30 1978
C
        COMMON /FLOCKC/ INTR
C      INTR 0= NO LOCK PERFORMED IN FDIT
C       INTR=1 LOCK PERFORMED IN FDIT

C      NOV 7. 1977
       INTEGER FDITF.FDS.FBDS
       LOGICAL FDBAK
       COMMON /FDITC/ FDITF(4).FDS.JTOTY(5.15).FBDS.FDBAK.IFDIT
       DIMENSION JTAUG(5).JTAUO(5).JTAUT(5).JTDRT(5).JTERT(5).
      1          JTERC(5).JTERL(5).JTERS(5).JTFLG(5).JTIX (5).
      2          JTLC (5).JTPUL(5).JTABC(5).JTEXT(5).JTBAK(5)
       EQUIVALENCE (JTAUG(1).JTOTY(1. 1)).(JTAUO(1).JTCTY(1. 2))
       EQUIVALENCE (JTAUT(1).JTOTY(1. 3)).(JTDRT(1).JTCTY(1. 4))
       EQUIVALENCE (JTERT(1).JTOTY(1. 5)).(JTERC(1).JTCTY(1. 6))
       EQUIVALENCE (JTERL(1).JTOTY(1. 7)).(JTERS(1).JTCTY(1. 8))
       EQUIVALENCE (JTFLG(1).JTOTY(1. 9)).(JTIX (1).JTCTY(1.10))
       EQUIVALENCE (JTLC (1).JTOTY(1.11)).(JTPUL(1).JTCTY(1.12))
       EQUIVALENCE (JTABC(1).JTOTY(1.13)).(JTEXT(1).JTCTY(1.14))
       EQUIVALENCE (JTBAK(1).JTOTY(1.15))
C      FDITF IS AN ARRAY CONTAINING THE DOCUMENT-NAME OF THE DOCUMENT
C                BEING EDITED. IN THE FIRST 3 WORDS. AND ALL BLANKS IN
C                THE 4TH WORD
C      FDS IS THE DATASET NUMBER OF THE DATASET CONTAINING THE DOCUMENT
C                BEING EDITED
C      FBDS IS THE DATASET NUMBER OF THE DOCUMENT COMMAND BACKUP DATASET
C      JTOTY IS 15 DIFFERENT ARRAYS. EACH 5 WORDS LONG - ONE FOR EACH OF
C                THE 14 DIFFERENT DOCUMENT TABLES. AND ONE FOR THE BACKUP
C                TABLE. FOR EACH ARRAY. THE 5 WORDS HAVE THE FOLLOWING
C                INFORMATION-
C                1. THE PARTICULAR PART OF THE FILE-NAME (FILNM(4))
C                2. THE ID OF THE FIRST RECORD
```

```
C      NDA    - MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C     [illegible]
C                  2. THE ID OF THE FIRST RECORD
C                  3. SIZE OF PRIMARY KEY
C                  4. SIZE OF SECONDARY KEY
C                  5. STORAGE MODE (PACK OR/AND POINT)
C      IFDIT IS 1 IF WE ARE IN FDIT-MODE; 0 OTHERWISE.
C
C


C      DOCUMENT KEYWORD INDEX TABLE COMMON
C      MAY 25, 1978
       INTEGER DKYWDT
       COMMON /FKYWDC/ DKYWDT(5,6),KWUSE
C      DKYWDT: DOCUMENT KEYWORD TABLE
C     DKYWDT(1,N)= THE PARTICULAR PART OF THE FILE-NAME
C     DKYWDT(2,N)= THE ID  OF THE FIRST RECORD
C     DKYWDT(3,N)= THE SIZE OF THE PRIMARY KEY
C     DKYWDT(4,N)= THE SIZE OF THE SECONDARY KEY
C     DKYWDT(5,N)= THE STORAGE MODE (PACK/POINT)
C      WHERE N IS THE KEYWORD INDEX (1 TO 6)
C      KWUSE=NUMBER OF KEYWORD INDEX TABLES USED BY THE DOCUMENT
C

C
C
C      DECEMBER 15, 1977
       COMMON /FLSLC/ IFLSW,IFLNO,IFLCH
C      IFLSW = 0 NORMAL CALL OF FLGCH
C      IFLSW = 1 FLGCH CALLED BY TEXTGC AND HENCE REQUIRES THE VALUES
C                OF IFLNO & IFLCH TO BE SPECIFIED VIA COMMON AND NOT VIA STRING
C      IFLNO THE FLAG ID
C      IFLCH THE FLAG CHOICE NO OF THE IFLNO FLAG ID
C
C

C JULY 22,1977
C
C
C
C
C
C
C
         COMMON /FORMAT/
     1              PFMTT( 15), PFMTB( 15),PFMS( 29),PFMNS( 12)
     1,PFMHF(12)          ,PFMTC(15)
C
C
C
           INTEGER  PFMTT,PFMTB,PFMS,PFMNS,PFMHF ,PFMTC
C
C
C
C
C
C
C          PFMTT(15)- PRINT FORMAT FOR THE MARGIN AT THE TOP OF PAGE
C          PFMTB(15)- PRINT FORMAT FOR THE MARGIN AT THE BOTTOM OF PAGE
C          PFMS (29)- PRINT FORMAT FOR SKIPPING LINES  - ALL COLUMNS
C          PFMNS(12)- PRINT FORMAT FOR NONSKIP OF LINES- TEXT ONLY
C          PFMHF(12)- PRINT FORMAT FOR HEADER,FOOTER,PAGENUMBER
C         PFMTC(15) - PRINT FORMAT FOR PAGE EJECT ON TERMINALS
C
C
C
C
```

```
C
C         MAY 4, 1978
C       GENERATE-UPDATE COMMON
        INTEGER FDOCN,SDOCN     ,SONDSN
        COMMON /GEDAC/ FDOCN(3),SDOCN(3),IOP ,SONDSN(2)
C       FDOCN IS FATHER DOCUMENT NAME.
C       SDOCN IS SON DOCUMENT NAME.
C         IOP = SWITCH TO INDICATE GENERATION OR UPDATING
C             = 1 :  GENERATION
C             = 2 :  UPDATING
C
C


        INTEGER GEN
        COMMON /GEUPC/                             IPULL,IPULLC(60,500),IPB
     1,GEN(125),LNXT,NFTOT(2),NUM(100,2),LINID,LINENO
C     IPULL    NUMBER OF PULL COMMANDS IN IPULLC ARRAY
C     IPULLC   ALL VALID PULL CONDITIONS
C       GEN(125) : STRING GENERATED IN EDITSPEC FORM
C       LNXT : LAST CHAR POSITION IN GEN
C       NFTOT(2) : COUNTERS FOR TOTAL NUMBERS OF COLUMNS AND ROWS.
C       NUM(100,2): STORAGE FOR HOLDING COLUMN NUMBERS AND ROW NUMBERS
C       LINID : ID USED IN C OR R TYPE PULL, ALSO USED AS A SWITCH TO
C                 INDICATE A CONTINUATION FROM A PREVIOUS C OR R PULL
C       LINENO : LINE NUMBER ENTERED IN SON'S DOCUMENT

C           OCTOBER 31,1978
C THIS COMMON IS USED TO TRANSMIT THE REQUIRED DATASET ACCESS FOR DYNAMIC ALLOC.
            COMMON  /INACTC/  IACCE
C
C VALUES FOR IACCE ARE:
C                            0 OR 1 = SHARED ACCESS REQUIRED
C                            2 = EXCLUSIVE ACCESS REQUIRED
C                            3 = CALLED TO CREATE A NEW  DATA SET


C
C         JULY 22, 1977
C       I/O UNIT NUMBERS
        COMMON /IOC/ INP1,INP2,NLST,NPRO,NECO,NPRI,MESG
C       INP1 IS PRIMARY INPUT LOGICAL UNIT NUMBER (OR LUN FOR SHORT)
C       INP2 IS SECONDARY INPUT LUN
C       NLST IS LISTING LUN (FOR COMMANDS .LIST OR .LT)
C       NPRO IS ON-LINE PRONPTING LUN
C       NECO IS LUN FOR ECHOING COMMANDS
C       NPRI IS PRINTING LUN (FOR .PR COMMAND)
C       MESG IS LUN FOR ALL MESSAGES AND ERRORS


C

C
C
C         NOVEMBER 8, 1978
C
C         RESTORE - COMMON, KEYWORD (SYSTEM) PARTITION
C
C

          COMMON     /KINDC/   CDBUF, INDEX, TYPE, PRVTYP, DOCCNT,
     *                         EXDOC, EXSEQ, SEQNM, KFERR, DOCBUF,
     *                         WRKBUF, SUBTBL, KNOTBL, LVORID, KACRID,
     *                         KYSIZE, KYWNO
C
C
                      INTEGER    CDBUF(65)
C                                          BUFFER TO HOLD VARIABLE PORTION
C                                          OF CONVERTION RECORDS
C       NDA  = MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
```

```
C       [illegible]
C       [illegible]
C                      2. THE ID OF THE FIRST RECORD
C                                                  OF CONVENTION RECORDS
C
                           INTEGER    INDEX
C                                                KEYWORD INDEX # OF TABLE BEING
C                                                 CURRENTLY RESTORED
C
                           INTEGER    TYPE
C                                                 TYPE CODE OF CONVERTION RECORD
C                                                  JUST READ
C
                           INTEGER    PRVTYP
C                                                 TYPE CODE OF CONVERTION RECORD
C                                                  PREVIOUS TO CURRENT ONE
C
                           INTEGER    DOCCNT
C                                                 COUNT OF DOCUMENTS CURRENTLY
C                                                  RECORDED ON ACCESS LIST WE
C                                                  ARE NOW RESTORING
C
                           INTEGER    EXDOC
C                                                 NUMBER OF DOCUMENT NAMES WE
C                                                  EXPECT TO RESTORE TO THIS
C                                                  ACCESS LIST
C
                           INTEGER    EXSEQ
C                                                 SEQUENCE NUMBER WE EXPECT THE
C                                                  NEXT RESTORE RECORD TO HAVE.
C
                           INTEGER    SEQNM
C                                                 SEQUENCE NUMBER OF RESTORE
C                                                  RECORD JUST READ.
C
                           LOGICAL    KFERR
C                                                 "FATAL ERROR" SWITCH DURING
C                                                  KEYWORD RESTORE PHASE
C
                           INTEGER    DOCBUF(4)
C                                                 BUFFER TO HOLD DOCUMENT NAMES AND
C                                                  FLAGS BEFORE APPENDING THEM TO
C                                                  THE DOCUMENT-ACCESS LISTS
C
                           INTEGER    WRKBUF(20)
C                                                 WORK BUFFER
C
                           INTEGER    SUBTBL(5)
C                                                 TABLE-IDENTIFIER FOR INDIVIDUAL
C                                                  KEYWORD SUB-TABLES
C
                           INTEGER    KNOTBL(5)
                                                  TABLE-IDENTIFIER FOR KEYWORD
C                                                  NUMBER TABLE.
C
                           INTEGER    LV0RID
C                                                 RECORD-ID OF LEVEL 0 RECORD (WHICH
C                                                  CONTAINS CRE-ID,KW-TBL POINTER,
C                                                  ETC.)
C
                           INTEGER    KACRID
C                                                 RECORD-ID OF LAST KEYWORD ACCESS
C                                                  RECORD GENERATED
C
                           INTEGER    KYSIZE
C                                                 SIZE OF CURRENT KEYWORD(S) IN
C                                                  CHARACTERS
C
                           INTEGER    KYWNO
```

```
                                    NETWORK NUMBER OF NETWORK JUST
C                                           RESTORED
C
C         - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -.
C
C
C
C      MARCH 6.1978
         INTEGER RSRCS
         COMMON /LOCKC/ RSRCS(7.20).NRSRCS
C
C  RSRCS(1.ALL) DATA SET NUMBER
C  RSRCS(2.ALL)      SYSTMTABLE WORD OR FIRST WORD OF DOCUMENT NAME
C  RSRCS(3.ALL)      SYSTMTABLE WORD OR 2 ND  WORD OF DOCUMENT NAME
C  RSRCS(4.ALL)      SYSTMTABLE WORD OR 3 RD  WORD OF DOCUMENT NAME
C  RSRCS(5.ALL)      SYSTMTABLE WORD OR 4 TH  WORD OF DOCUMENT NAME
C  RSRCS(6.ALL)  DATA BASE NUMBER (STARTING FROM 0).
C  RSRCS(7.ALL)  SHAIRED OR EXCLUSIVE USE REQUEST
C  NRSRCS  NUMBER OF RESOURCES REQUESTED
C
C


C
C
C         JULY 22. 1977
C      LOG-ON AND LOG-OFF COMMON
       INTEGER ACTNO.TLOG.CLTIM
       COMMON /LOGC/ ACTNO(3).TLOG(5).CLTIM
C
C      ACTNO IS ACCOUNT NUMBER CURRENTLY LOGGED-ON UNDER.
C      TLOG CONTAINS THE TIME AND DATE OF LOG-ON.

C
C LIST/PRINT COMMON USED TO PRODUCE A HARD COPY TEXT FOR EDITOR/ENGR
C
          COMMON /LTPTC/ LTPT
C
C  LTPT SWITCH 0= OFF,1= ON
C
C


C
C
C         JULY 22. 1977
C      AUXILIARY MACHINE SPECIFIC CONSTANTS
       COMMON /MC/ MAXCH.IWD1.NCUM.IWD2.NCUM2.NCU2.IWD3.NCU243
C      THIS COMMON BLOCK CONSISTS OF VALUES THAT COULD HAVE BEEN
C      COMPUTED FROM NCU (NUMBER OF CHARACTERS PER STANDARD-UNIT),
C      AND THAT ARE PRE-COMPUTED AND STORED IN COMMON TO SAVE TIME
C      MAXCH  = NCU*80+1
C      IWD1   = 1
C      NCUM   = NCU-1
C      IWD2   = NCU+1
C      NCUM2  = NCU*2-1
C      NCU2   = NCU*2
C      IWD3   = NCU*2+1
C      NCU243 = NCU*243
C
C

C JULY 22.1977
C
C
C
C
C
       COMMON /NUMBRC/ IPAGNO(15,1), NCPGNO, PAGE1, PAJ
C      DKYWNT(6) IS THE SIZE OF THE SECONDARY KEY
```

```
C   [illegible] THE SIZE OF THE SECONDARY KEY
C                  2. THE ID OF THE FIRST RECORD
C                                            OF CONVERTION RECORDS
      COMMON /NUMBRC/ IPAGNO(15),NCPGNO, PAGE( 96)
     1,                  NPNFC, IPNFC(96, 6)
     1, INPNFC(72,6),INPAGE(72)
     1,IPNFCC
C
       INTEGER PAGE, PAGENO
C
          DIMENSION  PAGENO(60),ISFL(6),ISTN(6),ISTA(6),ISLI(6),ISLP(6)
C
          EQUIVALENCE (PAGE( 1),ISFI  ),(PAGE( 8),LOP   ),(PAGE( 9),ISOP)
     1                ,(PAGE(10),IPJ   ),(PAGE(35),ITLP  )
     2                ,(PAGE( 2),ISFL(1)),(PAGE(11),ISTN(1))
     3                ,(PAGE(17),ISTA(1)),(PAGE(23),ISLI(1))
     4                ,(PAGE(29),ISLP(1)),(PAGE(37),PAGENO(1))
     5,(PAGE(36),NSD)
C
C
C
C
C   NUMBRC   COMMON  DESCRIPTION
C
C            IPAGNO(15)    -THE PAGE NUMBER IN MACHINE CHARACTERS ,
C                           PACKED FORMAT.
C            NCPGNO        -THE NUMBER OF WORDS IN (IPAGNO(15))
C            PAGE(96 )     -INFORMATION ABOUT THE PAGE NUMBER.
C        EQUIVALENCED AS FOLLOWS:
C        ISFI=      SUBFIELD TO INCREMENT
C        ISFL=      SUBFIELD LENGTH 1-6 SUBFIELDS
C    THE FOLLOWING READ FROM PAGE NUMBER FORMAT RECORD
C        LOP =      LOCATION ON PAGE
C        ISOP=      START ON PAGE NUMBER
C        IPJ =      PAGE JUSTIFICATION
C        ISTN=      SUBFIELD TYPE NUMERIC
C        ISTA=      SUBFIELD TYPE ALPHABETIC
C        ISLI=      SUBFIELD LAST CHARACTER OF INCREMENT
C        ISLP=      SUBFIELD LAST CHARACTER TO PRINT
C        ITLP=      TOTAL LENGTH OF PAGE NUMBER
C    NSD  NUMBER OF SUBFIELDS DEFINED
C        PAGENO=    ACTUAL PAGE NUMBER IN MACHINE CHARACTER
C                   UNPACKED FORM(RIGHT JUSTFIED)
C    NPNFC        =  NO OF PARAGRAPH NUMBER FORMATS IN COMMON
C            IPNFC( 96,6) - SIX DIFFERENT PARAGRAPH NUMBERS
C                         - INFORMATION IN SAME ORDER AS DEFINED
C                         - FOR (PAGE )
C        INPNFC = INITIALIZATION FOR PARAGRAPH NOS
C        INPAGE = INITIALIZATION FOR PAGE NOS
C        IPNFC     PAGE NUMBER FORMAT IN COMMON
C
C
C
C
C JULY 22,1977
C
C
C
C
C
C
C
          COMMON  /PAGEBF/ FLAG( 64),LNNUM( 64),PGBUF(40,64)
     1,                    AUDDOC(3,64),AUDTCN(64),TEXTSG( 64)
     3,                    IHEADO(40,12 ),IHEADE(40,12 ),HDLNO,HDLNE
     4,                    IFOOTO(40,12 ),IFOOTE(40,12 ),FTLNO,FTLNE
     5,                    NPFC,IPFT(4,12)
     6,                   MLBODY,KLN
     4,IHEADS(40),IHEAO,IHEAE,IHC,  IFOOTS(40),IFOOO,IFOOE,IFC
     [illegible]
     1,IPGCNT,ILRJ,NLCFP
C
C
C
C
           INTEGER  FLAG,PGBUF,AUDDOC,AUDTCN,TEXTSG
     1,             HDLNO,HDLNE,FTLNO,FTLNE
C
C
C
C
C    PAGEBF   COMMON  DESCRIPTION
C
C              FLAG(64) - THE FIRST FLAG FOUND ON THIS ROW ,NEGATIVE IF
C                       - MORE FLAGS ARE ON THIS ROW.
C              LNNUM(64)- THE LINE NUMBER IN THIS DOCUMENT OR THE NEGATIVE
C                       - OF THE LINE NUMBER COPIED FROM ANOTHER DOCUMENT.
C              PGBUF(40,64)- THE TEXT COLUMN FOR ONE PAGE
C              AUDDOC(3,64)- NAME OF THE DOCUMENT THAT THIS LINE HAS
C                          - BEEN COPIED FROM.
C              AUDTCN(64)  - CYCLE NUMBERSFOR DOCUMENT COPIED AND FOR
C                          - TEXT ENTRANCE INTO CURRENT DOCUMENT
C              TEXTSG(64)  - THE TEXT SEGMENT FOR THIS LINE
C      IHEADO = ODD PAGE HEADER STORAGE
C      IHEADE = EVEN PAGE HEADER STORAGE
C        HDLNO = NO LINES IN HEADER -ODD
C        HDLNE = NO LINES IN HEADER -EVEN
C      IFOOTO = ODD PAGE FOOTER STORAGE
C      IFOOTE = EVEN PAGE FOOTER STORAGE
C             FTLNO- NO. LINES IN FOOTER -ODD
C             FTLNE- NO. LINES IN FOOTER -EVEN
C             NPFC - NO. OF PARAGRAPH FORMATS LOADED INTO COMMON
C     IPFT(4,12) =PARAGRAPH FORMAT TABLE
C     IPFT(1, ALL) =  LEFT INDENTION OF FIRST LINE
C     IPFT(2, ALL) =  LEFT INDENTION ALL OTHER LINES
C     IPFT(3, ALL) =  RIGHT INDENTION OF ALL LINES
C     IPFT(4, ALL) =  PARAGRAPH ID
C        MLBODY = MAXIMUM NO. OF PRINTABLE LINES IN PGBUF FOR BODY OF TEXT
C        KLN    = LINE NUMBER IN PGBUF BEING WRITTEN. NEG. NOT FULL YET.
C        IHEADS(40) = HEADER STORAGE
C        IHEAE      = LINE LOCATION OF HEADER STORAGE -EVEN
C        IHEAO      = LINE LOCATION OF HEADER STORAGE -ODD
C        IHC        = HEADER STORAGE CHANGED 1= YES  0= NO
C        IFOOTS(40) = FOOTER STORAGE
C        IFOOE      = LINELOCATION OF FOOTER STORAGE - EVEN
C        IFOOO      = LINE LOCATION OF FOOTER STORAGE - ODD
C        IFC        = FOOTER STORAGE CHANGED 2= YES  0= NO
C      IOJPN = OUTSIDE JUSTIFICATION OF PAGE INFORMATION
C      + INDICATES AN EVEN PAGE
C      - INDICATES AN ODD PAGE
C      LFL = LAST FOOTNOTE LINE
C      LMAX = TOTAL NUMBER OF LINES ALLOWED ON THIS PAGE(MAX-FN)
C      IPFSN = PRINT FORMAT STATEMENT NUMBER IDENTIFIER
C         IPGCNT    NUMBER OF PAGES PRINTED IN THIS PRINT COMMAND
C         ILRJ      1= JEFT/RIGHT JUSTIFICATION OF EACH LINE BEFORE PRNT
C         NLCFP     NO. LINES TO FOLD/TERMINAL TYPEWRITER ONLY
C
C
C
C
C      AUGUST 29, 1977
C      READ AND PARSE COMMAND ROUTINES COMMON AREA.
       INTEGER PMPTR,STRNG,EDLIN,EDSIZ
       LOGICAL EOFIL,NXTRD,CMDOK
       COMMON /PARSC/MCLIN(400),STRNG(400),PMPTR(100,3),NXTLN(80),
     1                NCHAR,ISUB,EOFIL,NXTRD,CMDOK,EDLIN(400),MCSIZ,EDSIZ
C      MCLIN IS USED TO STORE LINE READ IN MACHINE FORMAT (A1) FOR WRITE.
C       NDA    - MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C      DKYWDTI4.NI = THE SIZE OF THE SECONDARY KEY
```

```
C               2. THE ID OF THE FIRST RECORD
C      [illegible]
C      STRNG CONTAINS THE CONVERTED COMMAND (IN EDITSPEC FORMAT)
C      PMPTR IS USED TO BREAK UP A COMMAND STRING INTO FIELDS OR PARAMETERS
C                PMPTR(,1) IS THE CHARACTER INDEX INTO THE ARRAY STRNG
C                          INDICATING START OF THE FIELD
C                PMPTR(,2) IS LENGTH OF FIELD IN CHARACTERS
C                PMPTR(,3) IS FIELD TYPE CODE
C      NXTLN IS THE CONTENTS OF THE NEXT LINE (NEXT COMMAND) IF IT HAS
C                 ALREADY BEEN INADVERTANTLY READ
C      NCHAR IS NUMBER OF CHARACTERS OF COMMAND IN ARRAY STRNG.
C      ISUB  IS COMMAND NUMBER CODE RETURNED BY COMMAND DECODER
C      EOFIL IS END-OF-FILE INDICATOR FOR READING
C      NXTRD IS .FALSE. IF NXTLN HAS NOT BEEN READ
C               .TRUE.  IF NEXT LINE HAS BEEN READ INTO ARRAY NXTLN
C      CMDOK IS .FALSE. IF COMMAND COULD NOT BE EXECUTED DUE TO ERROR
C               .TRUE.  IF COMMAND WAS EXECUTED AND MAY BE BACKED-UP
C      EDLIN STORES LINE READ IN PACKED EDITSPEC FORMAT.
C      MCSIZ IS NUMBER OF CHARACTERS READ INTO MCLIN.
C      EDSIZ IS NUMBER OF CHARACTERS PLACED INTO EDLIN.
C
C

C
            COMMON /PRERRC / IPRERR
C FOUND ONLY IN GTCMD AND DEBOG FOR SUPRESSION OF NOT A COMMAND PRINTING
C

C
C   AUGUST 1.1977
C   MODIFIED JULY 10, 1979:  CMBUF 126 CHANGED TO 400.
C
C
      COMMON /PRINTC/ DFDRC(58)
     1.                    FLGOP(6 ).IRSLN,IRELN.ISEG(3),NITC.LCO,NOD,NPPNI
     1, ICNTD.ICNOD.IODNM(3)
     1.INSTR(126).CMBUF(400).LPBUF(126)
     1,MORLN.MORIL.NCHCO.LSTRT.KSTRT.LINE,CTBP.LENG.ITAB1.ITAB2
     1, RID1.RID2.IHTEXT(243).IFLAGT.NOLIN.RTAB
     1, NPSW.K2PSW.KSW.LWID.SLEN.INDIL .INDLT.INDRT
     1, NEWST(126),MINSTR.MNEWST.LINSTR , LNEWST.NINSTR,NNEWST
     1.XBLNK.LWIDSU,IOFSET
     1.IPERID
     1.ILNCLN
     1.IUNSC
     1.LFLAG,INLINE.LAUD(3).ICYCLE.ITXTSG
     1.NSLVL(15),LSTCT(15).LENCT(15),ITOPS.LEVHST.INDST.LSNEW(15)
     1,LENEW(15)
C
C
      DIMENSION IPP(4,10)
    INTEGER CMBUF.CTBP.RTAB.XBLNK .SLEN,FLGOP
      INTEGER DFDRC
     INTEGER  RID2,RID1, TEXTDR(126)
C
C
     EQUIVALENCE  (DFDRC( 1).IFOOT ),(DFDRC( 2).IHEAD )
     1,(DFDRC( 3).ILNG  ),(DFDRC( 4).IWID  ).(DFDRC( 5).ILEVM )
     2,(DFDRC( 6).IPPNF ),(DFDRC( 7).IPPBP ).(DFDRC( 8).IPPBP1)
     3,(DFDRC( 9).IPPBP2).(DFDRC(10).IPNF  ),(DFDRC(11).INS   )
     4,(DFDRC(12),IRLJ  ),(DFDRC(13).MT    ).(DFDRC(14).MB    )
     5,(DFDRC(15).ISAF  ).(DFDRC(16).ISALN ),(DFDRC(17),ISAT  )
     6,(DFDRC(18).ISAA  ).(DFDRC(19).IPP(1.1))
     1,(LPBUF(110).LINES ),(LPBUF(111).LNSTA ),(LPBUF(112),LNEND )
     2,(LPBUF(113).LSFID ).(LPBUF(114).LNSPC ),(LPBUF(115),NLINE )
     3,(LPBUF(116).X1    ),(LPBUF(117).JUST  ),(LPBUF(118),X2    )
     4,(LPBUF(119).X3    ),(LPBUF(120).LENGX ),(LPBUF(121),X5    )
     6,(LPBUF(125).LENGE ).(LPBUF(126).NCHCE )
       EQUIVALENCE  ( TEXTOR(1),ICNTD)
C
C
C              DFDRC(48) - ALL INFORMATION RELATED TO THE DOCUMENT FORMAT
C            EQUIVALENCED AS FOLLOWS:
C  THE FOLLOWING ARE VARIABLES IN THE DOCUMENT FORMAT RECORD
C          IFOOT    = FOOTER FORMAT ID
C          IHEAD    = HEADER FORMAT ID
C          ILNG     = PAGE LENGTH
C          IWID     = PAGE WIDTH
C          ILEVM    = MAXIMUM NUMBER OF LEVELS
C          IPPNF    = PARAGRAPH NUMBER FORMAT
C          IPPBP    = PRINT PARG NO AT BOTTOM OF PAGE
C          IPPBP1   = PRINT PARG NO AT BOTTOM OF PAGE - 1 ST SUBFIELD
C          IPPBP2   = PRINT PARG NO AT BOTTOM OF PAGE - LAST SUBFIELD
C          IPNF     = PAGE NUMBER FORMAT ID
C          INS      = NO SKIP BETWEEN HEADER/FOOTER AND PAGE NUMBER
C          IRLJ     = RIGHT / LEFT JUSTIFICATION
C          MT       = TOP MARGIN INLINES
C          MB       = BOTTOM MARGIN IN LINES
C          ISAF     = SPACE BETWEEN  FLAGS AND LINE NUMBER COLUMNS
C          ISALN    = SPACE BETWEEN LINE NO AND TEXT COLUMNS
C          ISAT     = SPACE BETWEEN TEXT AND AUDIT TRAIL
C          ISAA     = SPACE BETWEEN AUDIT NAME AND CYCLE NOS
C          IPP(1,A) = PARAG SUBFIELD TO INCREMENT
C          IPP(2,A) = PARAG SUBFIELD TO PRINT FIRST
C          IPP(3,A) = PARAG SUBFIELD TO PRINT LAST
C          IPP(4,A) = PARAG FORMAT IDENT
C          FLGOP(6)= DATA COLUMNS TO BE PRINTED: 1- X-50     ,2- 38-L
C                    3-32-F               4-46-T              5-27-A  6-42-P
C          IRSLN- SLN REC ID
C            IRELN- ELN REC ID
C          ISLN= STARTING LINE NO   TO PRINT.USER INPUT
C          IELN= ENDING LINE NUMBER TO PRINT.USER INPUT
C          ISEG=      TEXT SEGMENT 1 THR3
C          ISPAC= LINE SPACING01=SINGLE 2= DOUBLE
C          NITC/ NO INDEX/TABLE CONTENTS
C          LCO/ LOGIC CHECK OVERRIDE
C          NOD-NUMBER OF OUTPUT DEVICE 0 OR 1
C     NPPNI- NO P/P INITIALIZATION
C            IFORM= DOCUMENT FORMAT ID
C          ICNTD  = CYCLE NUMBER FOR THIS LINE IN THIS DOCUMENT
C          ICNOD  = CYCLE NUMBER FOR THIS LINE IN OTHER DOCUMENT
C          IODNM(3)= OTHER DOCUMENT NAME
C          INSTR  = ONE LINE OF INPUT FROM A DOCUMENT TEXT TABLE
C          CMBUF  = THE ARRAY THAT CONTAINS THE COMMAND NEST
C          LPBUF  = 1-109 WORDS HOLD THE CURRENT LINE TO BE PRINTED  IN
C                   MACHINE  CHARACTERS
C          LINES  = TOTAL NUMBER OF LINES TO PRINT IN THE TEXT COLUMN
C          LNSTA  = STARTING LINE NUMBER TO PRINT
C          LNEND  = ENDING   LINE NUMBER TO PRINT
C          LSFID  = DOCUMENT FORMAT ID
C          LNSPC  = LINE SPACING FOR PRINTING
C          NLINE  = CURRENT LINE NUMBER
C          JUST   = TEXT JUSTIFICATION
C          LD     = NUAMBER OF CAPITALLETTERS FOUND IN ARRAY INSTR
C          LENGE  = NUMBER OF MACHINE CHARACTERS IN ARRAY  INSTR  BEFORE
C                   THE COMMAND STRING
C          NCHCE  = NUMBER OF CHARACTERS IN COMMAND STRING (MACHINE  CTRS)
C          MORLN  = NUMBER OF CHARACTERS(MACHINE) IN LPBUF
C          MORIL  = THERE ARE MORE CHARACTERS IN THIS LINE TO BE CHECKED
C                   IF EQUIL TO  ONE , NO MORE IN LINE IF EQUIL TO ZERO
C          NCHCO  = NUMBER OF CHARACTERS IN COMMAND STRING (EDITSPED CTRS)
C          LSTRT  = FIRST WORD IN ARRAY INSTR THAT HAS NOT BEEN SEARCHED
C                   FOR AN INTERNAL COMMAND. USED IN LOCAC
C     NDA    = MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C     NKYWDT(4,N)= THE SIZE OF THE SECONDARY KEY
```

```
C     [illegible]
C                    2. THE ID OF THE FIRST RECORD
C     [illegible]
C      KSTRT  = FIRST CHARACTER IN FIRST WORD IN ARRAY INSTR THAT HAS
C                 NOT BEEN SEARCHED FOR AN INTERNAL COMMAND. USED IN
C                 LOCAC
C      LINE   = INSTRUCTIONS FOR PROCESSING A LINE :NEG- COPY COMMAND
C                 ZERO- THIS DOCUMENT .POSITIVE- CONTINUE TO SCAN PRESENT
C                 LINE FOR COMMAND.
C      CTBP   = COMMAND TO BE PROCESSED IF EQUIL TO ONE,NONE IF ZERO.
C      LENG   = NUMBER OF INTERNAL CHARACTERS IN ARRAY INSTR BEFORE
C                 THE COMMAND STRING
C      ITAB1  = LEFT MARGIN OF TEXT COLUMN
C      ITAB2  = RIGHT MARGIN OF THE TEXT COLUMN
C      RID1=  RECORD ID OF THE LINE TO BE READ NEXT
C      RID2= THE RECORD WHERE THE START OF THE TABLE IS FOUND
C      IHTEXT(243)= HEADER RECORD IN TEXT TABLE -CURRENT RECORD
C      IFLAGT     = POINTER TO THE LAST/NEXT TEXT TABLE KEY.
C      NOLIN = 1=NO MORE LINES TO BE PRINTED.=0.MORE LINES TO BE PRINTED
C      RTAB= +10  = TABLR ROUTINE SAID THE TABLE IS COMPLETE
C      RTAB= -2 = PROCESSING A  COPY TABLE COMMAND
C      RTAB= +2 = PROCESSING A  TB  COMMAND
C      RTAB=  0 = NORMAL LINE PROCESSING
C      NPSW =
C              -1  =  *P* NOT NEXT CTR TO BE PROCESSED
C              -3  =  * LJ,RJ,CJ * NOT NEXT
C               0  =
C               1  =  *P* COMMAND READY
C               3  =  *JUST* COMMAND REAY
C              -1  =  *SL* NO PREVIOUS *P*
C      K2PSW = 0 =
C              1 =  TWO *P* COMMANDS ON SAME LINE
C              1 = *SL* FOUND AFTER A *P*
C     KSW    =
C     LWID   =    NUMBER OF CTRS IN PRINT LINE
C     SLEN   = SPACE AT LINE END IN THE (KSW) CURRENT PRINT LINE
C     NEWST(126) = INPUT (INSTR) IN EXTERNAC-MACHINE CTRS
C     MINSTR = MAXIMUM WORDS ACTUALLY READ INTO INSTR
C     MNEWST = MAXIMUM WORDS ACTUALLY PLACED INTO NEWST
C     LINSTR = LAST CHARACTER PROCESSED IN INSTR
C     LNEWST = LAST CHARACTER PROCESSED IN NEWST
C     NINSTR = TOTAL CTRS ACTUALLY IN INSTR
C     NNEWST = TOTAL CTRS ACTUALLY IN NEWST
C     XBLNK = ONE MACHINE BLANK RIGHT JUST
C     LWIDSU = NO. SUS IN LINE WIDTH
C    IOFSET = NO. COMMAND CTRS TO SKIP TO REACH NEXT CTR TO PROCESS
C      INDRT=   RIGHT INDENTATION
C      INDLT=     LEFT INDENT
C      INDIL      =FIRST PARAGRAPH INDENT
C       IPERID    =ONE MACHINE PERIOD IN RIGHT LOCATION
C      ILNCLN     CLEAN UP BLANKS ON LINE:1=YES.0=NO
C      IUNSC      MACHINE UNDERSCORE CHARACTER
C
C      IFLAG      CURRENT FLAG ID
C      INLINE     CURRENT LINE NUMBER TO PRINT
C      IAUD(3)      CURRENT DOC NAME TO PRINT
C      ICYCLE     CURRENT CYCLE TO PRINT
C      ITXTSG     CURRENT TEXT SEGMENT TO PRINT
C     NSLVL = STACK HOLDING NESTED LEVEL NUMBER FOR INTERNAL COMMANDS.
C     LSTCT = STACK HOLDING LOCATIONS OF THE STARTING CHARACTER OF
C             NESTED INTERNAL COMMANDS.
C     LENCT = STACK HOLDING LOCATIONS OF THE ENDING CHARACTER (*)
C             OF NESTED INTERNAL COMMANDS.
C     ITOPS = POINTER POINTS TO THE TOP OF STACK. (NOTE THAT THE FIRST
C             ELEMENT OF ARRAY IS CONSIDERED AS THE BOTTOM OF THE
C             STACK. E.G. NSLVL(1) IS THE BOTTOM OF THE STACK IF ITOPS
C             .GT. 1)
C     LEVHST = HIGHEST NESTING LEVEL.
C     [illegible]
```

```
C        ERRS. = INDICATOR OF STATUS. SEE VARIABLE COMMENT.
C        LSNEW = STACK HOLDING STARTING LOCATION OF COMMANDS IN NEWST.
C        LENEW = STACK HOLDING ENDING LOCATION (*) OF COMMANDS IN NEWST.
C
C
C
C
C

C         SEPT 1979
          COMMON  /PRTBLC / ACTALN
C     ACTALN    *CT* COMMAND LINE NUMBER

C
C
C       .. RESTORE COMMON - DOCUMENT PARTITION
C
C       .. MARCH 25, 1979
C
C
          COMMON /RESDOC/      DOCNM, DSNAME, DSNO, TYPE, OLDTYP, SEQNM,
     *                         EXSEQN, CVREC, CVIN, RESERR, CREUID, RESBUF
C
C
                    INTEGER    DOCNM(3)
C                                        NAME (IN EDITSPEC CHARS) OF DOC
C                                         CURRENTLY BEING TRANSFERRED TO
C                                         THE CONVERTION MEDIUM
C
                    INTEGER    DSNAME(2)
C                                        DATASET NAME TO WHERE THE CURRENT
C                                         DOCUMENT IS TO BE STORED
C
                    INTEGER    DSNO
C                                        DATASET NUMBER OF THE DATASET
C                                         IDENTIFIED IN DSNAME()
C
                    INTEGER    TYPE
C                                        TYPE CODE OF RESTORE RECORD JUST
C                                         OBTAINED
C
                    INTEGER    OLDTYP
C                                        TYPE CODE OF THE LAST RESTORE
C                                         RECORD (USED TO ENFORCE SEQUENC
C                                         OF RESTORE OPERATIONS)
C
                    INTEGER    SEQNM
C                                        SEQUENCE NUMBER OF THE CURRENT
C                                         RESTORE RECORD
C
                    INTEGER    EXSEQN
C                                        (EXPECTED) SEQUENCE NUMBER OF THE
C                                         NEXT RESTORE RECORD TO BE READ
C
                    INTEGER    CVIN
C                                        LOGICAL UNIT NUMBER OF THE CONV
C                                        INPUT DATA SOURCE
C
                    LOGICAL    RESERR
C                                        SWITCH TO SHOW IF AN ERROR HAS
C                                         OCCURED DURING THE RESTORE PROCESS
C
                    INTEGER    CREUID(3)
C                                        CREATOR USER ID OF THE OWNER OF
C                                         THE CURRENT DOCUMENT
C
                    INTEGER    CVREC(70)
C       NDA   = MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C     DKYWDT(6,N)= THE SIZE OF THE SECONDARY KEY
```

```
C     [illegible]
C     [illegible] 2. THE ID OF THE FIRST RECORD [illegible]
C                                          RAW FORM OF THE CONVERTION RECORD
C                                           DATA. TO RE SUPPLIED TO THE
C                                           INDIVIDUAL RESTORE ROUTINES
C
                      INTEGER   RESBUF(100)
C                                          GENERAL PURPOSE BUFFER AREA FOR
C                                           THE RESTORE RCUTINES TO SHUTTLE
C                                           INFORMATION BETWEEN THEMSELVES.
C                                           THE ACTUAL FORMAT OF DATA IN THIS
C                                           BUFFER IS DEPENDENT ON THE
C                                           CLUSTER OF RESTORE ROUTINES
C                                           CURRENTLY ACTIVE (IE, RESTORING
C                                           TEXT TABLE, AUDIT TABLES, ETC)
C
C
C
C
C
C     AUG 17, 1977
C     SCRATCH COMMON, FOR QUANTITIES USED AS DUMMY ARRAYS IN DIFFERENT ROUTINES.
      COMMON /SCRTC/ IHED(243)
C     IHED(243) IS USED AS ARGUMENT TO TABLE-HANDLER ROUTINES.
C
C
C     SEARC ROUTINE COMMON
C     THIS COMMON AREA WAS ADDED TO REDUCE THE NUMBER OF PARAMETERS
C     PASSED TO SEARC TO A TOLERABLE LEVEL.
C     THE ROUTINES CALLING SEARC ARE:LOCAT,CHANG,ERASE,LDHAR
C
      INTEGER RITO,TSID,SW50
      COMMON /SEARCH/ LBLNK,LSBLN,LITO(50,4),LIPA(50)  ,RITO(50,2),
     *ILITO,ILIPA,IMODE,IBYT,NRMTCH,IASEN,NTEFO,TSID(10),SW50,
     *ICNT,ITST
C
C       LBLNK: LEADING BLANKS IN A SOUGHT STRING.
C       LSBLN: TRAILING BLANKS IN THE SOUGHT STRING.
C       LITO: ARRAY TO HOLDLINE NUMBERS AND POSITIONS OF STRING.
C             LITO(N,1):LINE NUMBER  OF LINE WHERE STRING STARTS.
C             LITO(N,2): POSITION IN LINE OF THE FIRST CHARACTER
C                        OF THE SOUGHT STRING..
C             LITO(N,3): LINE NUMBER OF LINE WHERE STRING ENDS.
C             LITO(N,4): POSITION IN LINE WHERE STRING ENDS.
C       LIPA: SAME AS LITO USED FOR NEAR MATCH FINDINGS.
C       RITO(50,2): RECORD ID'S OF STARTING AND ENDING LINE NUMBERS
C                   OF EACH OCCURANCE OF THE SOUGHT STRING.
C       ILITO: NUMBER OF EXACT OCCURANCES.
C       ILIPA: NUMBER OF NEAR MATCH OCCURANCES.
C       IMODE: SWITCH THAT INDICATES TO SEARC ROUTINE WHETHER TO
C              SEARCH FOR ALL OR ONLY ONE OCCURANCE OF THE SOUGHT
C                     STRING IN EACH AREA SPECIFIED.
C       NRMTCH: A SWITCH THAT IS SET TO ONE IF THE NEAR MATCH
C               OCCURANCES ARE TO BE LISTED.
C       IASEN: WHEN THIS SWITCH IS ZERO WE DECAPITALIZE THE STRING
C              LEAVE THE STRING EXACTLY AS IT IS.
C              (REMOVE ALL CENT SIGNS) BEFORE MATCHING. ELSE WE
C       NTEFO: NUMBER OF TEXT SEGMENT ID'S GIVEN BY USER.
C       TSID(10): ARRAY CONTAINING THE TEXT SEGMENT ID'S.SEARCHING
C              WILL OCCUR ONLY IN SPECIFIED LINES OR PAIRS OF LINES
C              AND THEN ONLY AMONG THOSE LINES THAT HAVE  ONE OF THE
C              TEXT SEGMENT ID'S IN THE TSID ARRAY.
C       SW50: SINCE THE NUMBER OF OCCURANCES OF A STRING IN AN AREA
C             IS LIMITED BY THE SIZES OF ARRAY LITO(TO 50) THIS
C             SWITCH WILL BE SET TO ONE IF THERE ARE MORE OCCURANCES
```

```
C
C
C         JULY 22. 1977
C       SIZE COMMON
        COMMON /SIZEC/ NUSID.NACTN.NDSNM.NDOCN.NDATE.NTIME
C       NUSID IS SIZE OF USER-ID (12 CHARACTERS) IN WORDS = (NCU+11)/NCU
C       NACTN IS SIZE OF ACCOUNT-NUMBER (12 CHARACTERS) IN WORDS
C       NDSNM IS SIZE OF DATASET NAME (6 CHARACTERS) IN WORDS.
C       NDOCN IS SIZE OF DOCUMENT NAME (12 CHARACTERS) IN WORDS.
C       NDATE IS SIZE OF DATE FIELD (DD-MMM-YY  9 CHARACTERS) IN WORDS.
C       NTIME IS SIZE OF TIME-OF-DAY (8 CHARACTERS) IN WORDS.
C
C

C AUGUST 29.1979
        COMMON /SKPGC/ ISP
C ISP  SKIP PAGE SWITCH =0 NOSKIP. =1 SKIP PAGE
C COMMON USED IN PRNTB AND STRME SUBROUTINES

C
C       SYSTEM KEYWORD INDEX TABLE COMMON
C       APRIL 13. 1978.
        COMMON /SKYWDC/KYWDT(5.6)
C       KYWDT : SYSTEM KEYWORD TABLE
C       KYWDT(1.N)= THE PARTICULAR PART OF THE FILE-NAME
C       KYWDT(2.N)= THE ID  OF THE FIRST RECORD
C       KYWDT(3.N)= THE SIZE OF THE PRIMARY KEY
C       KYWDT(4.N)= THE SIZE OF THE SECONDARY KEY
C       KYWDT(5.N)= THE STORAGE MODE (PACK/POINT)
C       WHERE N IS THE KEYWORD INDEX (1 TO 6)
C
C


C
C       AUG 17. 1977
C       SYSTEM COMMON
        INTEGER SDS.SBDS.SYSFL.BATCH.USID.EDTBL
        LOGICAL MULTI.BAKUP
        COMMON /SYSTM/ IBLNK.JBLNK.SDS.SBDS.SYSFL(4).USID(3).LOGGD.MULTI.
       1BAKUP.BATCH.ISQTY(5.13).ITNM(3.8).IGEN.ISUP.ISBDD(5).IDOC
       2.ICLEAR
        DIMENSION ISACS(5).ISCHA(5).ISDIR(5).ISSPC(5).ISUSR(5).
       1ISDOC(5).ISFOF(5).ISHEF(5).ISPGN(5).ISPRF(5).ISPRN(5).ISTTF(5).
       2ISBAK(5)
        EQUIVALENCE (ISACS(1).ISQTY(1. 1)).(ISCHA(1).ISQTY(1. 2))
        EQUIVALENCE (ISDIR(1).ISQTY(1. 3)).(ISSPC(1).ISQTY(1. 4))
        EQUIVALENCE (ISUSR(1).ISQTY(1. 5)).(ISDOC(1).ISQTY(1. 6))
        EQUIVALENCE (ISFOF(1).ISQTY(1. 7)).(ISHEF(1).ISQTY(1. 8))
        EQUIVALENCE (ISPGN(1).ISQTY(1. 9)).(ISPRF(1).ISQTY(1.10))
        EQUIVALENCE (ISPRN(1).ISQTY(1.11)).(ISTTF(1).ISQTY(1.12))
        EQUIVALENCE (ISBAK(1).ISQTY(1.13))
        EQUIVALENCE (EDTBL.IBLNK).(MCHBL,JBLNK)
C
C       IBLNK IS A WORD FULL OF EDITSPEC BLANKS (ALSO KNOWN AS EDTBL)
C       JBLNK IS A WORD FULL OF MACHINE BLANKS (ALSO KNOWN AS MCHBL)
C       SDS IS THE DATASET NUMBER OF THE SYSTEM DATASET.
C       SBDS IS THE DATASET NUMBER OF THE SYSTEM BACKUP DATASET
C       SYSFL(4) CONTAINS THE SYSTEM TABLE GENERIC FILE NAME IN ITS
C                 LAST 3 WORDS. THE FIRST WORD IS BLANK.
C                 (SYSFL(I).I=2.4) =/124 SYSTEM TABLE/
C       USID(3) IS THE CURRENT USER ID
C       LOGGD IS ZERO IF NOBODY HAS LOGGED-ON. IF A VALID USER LOGS ON.
C                 LOGGD CONTAINS THE RECORD ID OF THE DATA RECORD ASSOCIATED
C                 WITH THE USER IN THE USER TABLE.
C       MULTI IS .TRUE. IF THIS IS A MULTIPLE USER EDITSPEC SYSTEM.
C                 AND .FALSE. OTHERWISE. IF MULTI IS .FALSE.. A SINGLE
C       NUMBER OF DIRECT-ACCESS DATASETS
C       KYWDT(4.N)= THE SIZE OF THE SECONDARY KEY
```

```
C     [illegible]
C     [illegible] 2. THE ID OF THE FIRST RECORD [illegible]
C               USER EDITSPEC SYSTEM IS CREATED.
C     BAKUP IS .TRUE. IF BACK-UPS ARE TO BE KEPT, SO THAT THE SYSTEM
C               MAY BE RECREATED AFTER FAILURE.
C     BATCH IS ZERO IF THE EDITSPEC SYSTEM IS FOR USE IN INTERACTIVE MODE.
C               ONE IF IN BATCH MODE.
C     ISQTY IS ACTUALLY 13 DIFFERENT ARRAYS. FOR EACH OF THE 13 ARRAYS,
C               THE FOLLOWING 5 QUANTITIES ARE DEFINED, IN ORDER
C               1. THE PARTICULAR PART OF THE FILE-NAME (FILNM(1))
C               2. THE ID OF THE FIRST RECORD
C               3. THE SIZE OF THE PRIMARY KEY
C               4. THE SIZE OF THE SECONDARY KEY
C               5. THE STORAGE MODE (PACK/POINT)
C     FOR EACH FIVE WORD ARRAY, THE FORTRAN VARIABLE NAME, THE PARTICULAR
C               PART OF THE FILE-NAME (FILNM(1)), AND A DESCRIPTION OF THE
C               TABLE (FILE) FOLLOWS.
C
C
C     IS---    VALUE OF IS---(1)        DESCRIPTION
C                (FILNM(1))
C     -----------------------------------------------------------------
C
C
C     ISACS         'ACS'                ACCOUNTS TABLE
C     ISCHA         'CHA'                CHARGES TABLE
C     ISDIR         'DIR'                DOCUMENT DIRECTORY
C     ISSPC         'SPC'                SPECS TABLE
C     ISUSR         'USR'                USER TABLE
C     ISDOC         'DOC'                DOCUMENT FORMAT
C     ISFOF         'FOF'                FOOTR FORMAT
C     ISHEF         'HEF'                HEADER FORMAT
C     ISPGN         'PGN'                PAGE NUMBER FORMAT
C     ISPRF         'PRF'                PARAGRAPH FORMAT
C     ISPRN         'PRN'                PARAGRAPH NUMBER FORMAT
C     ISTTF         'TTF'                TEXT TABLE FORMAT
C     ISBAK         'BAK'                BACK-UP TABLE
C     -----    ----------------        ----------
C
C     ITNM IS USED TO PRINT ERROR MESSAGES BASED ON PMPTR CODE.
C     IGEN IS USUALLY ZERO; EXCEPT DURING GENERATE/UPDATE, WHEN IT IS ONE.
C     ISUP IS SUPERVISOR SWITCH.
C     ISBDD(5) IS TABLE DEFINITION ARRAY FOR ANOTHER SYSTEM TABLE.
C     IDOC IS FORMAT-ID OF LEVEL 1 DOCUMENT FORMAT COMMAND LAST PROCESSED.
C     ICLEAR IS 1 IF BUFFERS ARE TO BE CLEARED AFTER EVERY COMMAND.
C
C

C


C
C
C       AUG 5, 77
C     TABLE HANDLING ROUTINES COMMON AREA
      INTEGER HORID
      COMMON /TABLC/ NLOC,ISHDR(11),IRHDR(243),HORID
C     ISHDR,IRHDR ARE SCRATCH SUPER- AND REGULAR- HEADERS USED BY TBL-HNDLR.
C     HORID IS ID OF IHDR RETURNED BY TBLCS
C
C


C
C
C     SEPT 27, 1977
C     TABL AND COPY TABLE SWITCHES COMMON.
      INTEGER TBINIT,TBCTSW,TBERR
      COMMON /TBCTSC/ TBINIT,TBCTSW,TBERR
```

```
C     TBINIT IS 1 FOR INITIALIZING CALL ,  LATER 0.
C     TBCTSW IS 1 FOR *TB. AND 2 FOR *CT.
C     TBERR IS 0 FOR NO TABLE ERROR SO FAR, >0 FOR ERROR ENCOUNTERED.
C
C


C
C     SEPTEMBER 5,1978
        COMMON /TBERRC/ LNERRO
C LNERROR   LINE NUMBER JUST READ
C
C


C
C
C     OCTOBER 22, 1977
C   MODIFIED JULY 10,1979: ADD MAXCHR: CHARS(500) TO CHARS(1000).
      INTEGER CHARS,LEN
      COMMON /TBLKC/ CHARS(1000),LEN,MAXCHR
C
C     CHARS(1000) IS RETURNED BY STRME, AS A BLOCK, IN UNPACKED EDITSPEC.
C     LEN IS NUMBER OF CHARACTERS IN CHARS.
C
C


C
C
C     OCTOBER 22, 1977
C   MODIFIED JULY 10, 1979:  TBCHRS(485) TO TBCHRS(1000).
C     BLOCK, IN INTERNAL (EDITSPEC), COMMON.
      INTEGER TBCHRS(1000),TBLEN,TBC,TBCODE
      COMMON /TBLKIC/ TBCHRS,TBLEN,TBC,TBCODE
C
C     TBCHPS(1000) STORES A BLOCK OF CHARACTERS IN UNPACKED  EDITSPEC FORMAT.
C     TBLEN IS LENGTH OF CHARACTERS IN TBCHRS.
C     TBC IS INDEX OF LAST CHARACTER PROCESSED IN TBCHRS.
C     TBCODE IS POSITIVE, IF TBCHRS IS A TABLE-RELEVANT COMMAND:
C             1,2,3,4,5,6 CORRESPOND TO *TH, *TE, *TB, *R, *CT, *CO RESPECTIVELY.
C
C


C
C
C     FEB 17, 1978
      INTEGER RMSAV,PNSAV,TBCMX,TBRCH,RWFND,HDFND
      INTEGER PRES,PREL,SUFS,SUFL,DCOLS,ROWID,ROWID
      COMMON /TBNITC/ LMSAV,LMISAV,RMSAV,TBCMX,TBRCH,PNSAV,RWFND,HDFND,
     1PRES(40),PREL(40),SUFS(40),SUFS(40),DCOLS(40),ROWID(6),RMNDX(5)
C
C     LMSAV SAVES OLD VALUE OF LEFT MARGIN FOR PARAGRAPH.
C     LMISAV IS LEFT MARGIN FOR FIRST LINE OF PARAGRAPH.
C     RMSAV IS RIGHT MARGIN.
C     PNSAV IS PARAGRAPH NUMBERING FORMAT OF PREVIOUS *P COMMAND SAVED.
C     TBCMX IS MAXIMUM NUMBER OF CHARACTERS IN A TABLE ROW.
C     TBRCH IS A WORD PACKED WITH BETWEEN-ROW CHARACTERS IN MACHINE FORMAT.
C     RWFND IS 1 IF START OF NEXT ROW (OR TBL-HDR OR TBL-END) HAS BEEN READ.
C              0 IF TBREAD MUST BECALLED TO GET A NEW LINE.
C     HDFND IS INITIALLY 0, SET TO 1 IF A ROW RETURNED IS A TBL-HEADER.
C     PRES IS INDEX OF CHARACTER COLUMN IN LINE WHERE PREFIX FIELD STARTS
C     PREL IS LENGTH OF PREFIX:  INDEX IS TABLE COLUMN NUMBER.
C     SUFS IS INDEX WHERE SUFFIX STARTS.
C     SUFL IS LENGTH OF SUFFIX (IN CHARCTERS).
C     DCOLS IS POSITION OF DECIMAL POINT IN A FIELD.
C     ROWID(6) IS ROW-ID AT DIFFERENT LEVELS OF *CT.
C     RMNDX(5) IS USED FOR INDEXING TBPWM(2,,5) TO GET NEXT ROW-ID.
C           ROWID AND RMNDX ARE INITIALIZED TO ZERO BY TBNIT
C     NDX  = MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C   DKYWDT(6,N) IS THE SIZE OF THE SECONDARY KEY
```

```
C      [illegible]
C      [illegible]
C
C


C
C
C      OCTOBER 22, 1977
C      /TABLE-NOTE/ COMMON.
       INTEGER TNARR,TNLEN,TNNUM
       COMMON /TBNOTC/ TNARR(500),TNLEN,TNNUM
C
C      TNARR(500) IS USED TO STORE TABLE-NOTES, IN MACHINE A1.
C      TNLEN IS INDEX OF LAST WORD OF TNARR THAT HAS BEEN FILLED UP.
C      TNNUM IS LAST TABLE-NOTE IDENTIFICATION NUMBER USED SO FAR.
C
C


C
C
C      OCTOBER 22, 1977
       INTEGER TBNAM,TBBRKC(6),TBCLL(40,6),TBNCL(6),TBCLM(40,5)
       INTEGER TBRWM(2,100,5),TBNRW(5),TBLVL,TBDOC(3,5),TBLNO(5)
       INTEGER TBHDR(243),TBFLG,TBLOC
       COMMON /TBPARM/TBNAM,TBBRKC,TBCLL,TBNCL,TBRWM,TBNRW,TBLVL,TBDOC,
      1TBLNO,TBHDR,TBFLG,TBLOC
C
C      TBNAM IS TABLE-FORMAT-ID.
C      TBBRKC(6) IS BREAK CHARACTERS, AT VARIOUS LEVELS OF *CT.
C      TBCLL IS COLUMN-ID-LIST IN VARIOUS *TB COMMANDS.
C      TBNCL IS NUMBER OF ENTRIES IN TBCLL
C      TBRWM IS ROW MAP (LIST OF SOURCE-ID, DESTN-ID) AT VARIOUS LEVELS.
C      TBNRW IS NUMBER OF ENTRIES IN TBRWM
C      TBLVL IS CURRENT *CT LEVEL.
C      TBDOC SAVES DOCUMENT NAMES FOR *CT.
C      TBLNO SAVES THE LINE NUMBER OF THE *CT COMMANDS, FOR GOING BACK.
C      TBHDR, TBFLG, TBLOC ARE FOR TABLE-HANDLING ROUTINES.
C
C


C
C
C      16 MAY,1978
C      TABLE-PARAMETERS ACS-TABLE COMMON.
       INTEGER DR1ACS,DR2ACS,DR3ACS,DR4ACS
       COMMON /TPACSC/ DR1ACS(4),DR2ACS(9),DR3ACS(1),DR4ACS(30),ID1ACS,
      1ID2ACS,ID3ACS,ID4ACS,I2ACS3,I2ACS4,L1ACS,L2ACS,L3ACS,L4ACS
C      DRNACS() IS FOR STORING DATA RECORD N OF ACS-TABLE.
C      IDNACS IS RECORD-ID OF DATA RECORD N OF ACS-TABLE.
C      I2ACS3 IS INDEX OF ELEMENT 3 OF DATA RECORD 2 OF ACS-TABLE.
C      I2ACS4 IS INDEX OF ELEMENT 4 OF DATA RECORD 2 OF ACS-TABLE.
C      LNACS IS LENGTH OF DATA RECORD N OF ACS-TABLE (IN CHARACTERS).
C
C


C
C
C        JULY 22, 1977
C      DOCUMENT AUDIT TRAIL (GENERAL) TABLE PARAMETERS COMMON.
       INTEGER DR1AUG,DR2AUG
       COMMON /TPAUG/ DR1AUG(11),DR2AUG(240),ID1AUG,ID2AUG,I1AUG5,
      1I1AUG6,L1AUG,L2AUG
C
C      DR1AUG HOLDS DATA RECORD 1 FROM AUG-TABLE.
C      DR2AUG HOLDS DATA RECORD 2 FROM AUG-TABLE.
C      ID1AUG IS RECORD ID OF DR1AUG.
C      ID2AUG IS RECORD ID OF DR2AUG.
C      I1AUG5 IS INDEX OF ELEMENT 5 OF DR1AUG.
```

```
C      I1AUG6 IS INDEX OF ELEMENT 6 OF DR1AUG.
C      L1AUG IS LENGTH OF DR1AUG (IN CHARACTERS;.
C      L2AUG IS LENGTH OF DR2AUG (IN CHARACTERS).
C
C

C        JULY 22. 1977
C      TABLE PARAMETERS COMMON.  FOR DIR-TABLE.
       INTEGER DR1DIR.DR2DIR.DR3DIR
       COMMON /TPDIRC/ DR1DIR(15).DR2DIR(12),DR3DIR(28),ID1DIR.ID2DIR.
      1ID3DIR.I1DIR9.I1DIRA.L1DIR.L2DIR.L3DIR
C      DR1DIR HOLDS DATA RECORD 1 FROM DIR-TABLE.
C      DR2DIR HOLDS DATA RECORD 2 FROM DIR-TABLE.
C      DR3DIR HOLDS DATA RECORD 3 FROM DIR-TABLE.
C      ID1DIR IS ID OF DATA RECORD 1 FROM DIR-TABLE.
C      ID2DIR IS ID OF DATA RECORD 2 FROM DIR-TABLE.
C      ID3DIR IS ID OF DATA RECORD 3 FROM DIR-TABLE.
C      I1DIR9 IS INDEX OF ELEMENT 9 IN DATA RECORD 1 OF DIR-TABLE.
C      I1DIRA IS INDEX OF ELEMENT 10 IN DATA RECORD 1 OF DIR-TABLE (A=10).
C      L1DIR IS LENGTH. IN CHARACTERS. OF DATA RECORD 1 OF DIR-TABLE.
C      L2DIR IS LENGTH. IN CHARACTERS. OF DATA RECORD 2 OF DIR-TABLE.
C      L3DIR IS LENGTH. IN CHARACTERS. OF DATA RECORD 3 OF DIR-TABLE.
C
C


C
C
C       JULY 22. 1977
C      TABLE PARAMETERS COMMON.  FOR DRT-TABLE.
       INTEGER DR1DRT
       COMMON /TPDRTC/ DR1DRT(32).ID1DRT.L1DRT
C      DR1DRT HOLDS DATA RECORD 1 FROM DRT-TABLE.
C      ID1DRT IS ID OF DATA RECORD 1 FROM DRT-TABLE.
C      L1DRT IS LENGTH. IN CHARACTERS. OF DATA RECORD 1 OF DRT-TABLE.
C
C

C
C      TABLE PARAMETERS COMMON, FOR LC-TABLE.
       INTEGER DR1LC
       COMMON/TPLCC/ DR1LC(22).ID1LC.L1LC
C      DR1LC HOLDS DATA RECORD 1 FROM LC-TABLE.
C      ID1LC IS ID OF DATA RECORD 1 FROM LC-TABLE.
C      L1LC IS LENGTH. IN CHARACTERS. OF DATA RECORD 1 OF LC-TABLE.
C
C

C
C      DECEMBER 8. 1977
C      PUL-TABLE PARAMETERS COMMON.
       INTEGER DR1PUL
       COMMON /TPPULC/ DR1PUL(64).ID1PUL.I1PULD.L1PUL
C
C       DR1PUL(64) IS DATA RECORD 1 OF PUL-TABLE.
C      ID1PUL IS ID OF DATA RECORD 1 OF PUL-TABLE.
C       I1PULD IS INDEX OF ENTRY 13 IN DR1PUL
C      L1PUL IS LENGTH OF DR 1: =(I1PULD-1+NDOCN+#SPECS)*NCU
C
C


C
C
C      SEP 26. 1977
C      TABLE PARAMETER COMMON FOR TTF-TABLE.
       INTEGER DR1TTF.L1TTF
C      NDA    = MAXIMUM NUMBER OF DIRECT-ACCESS DATASETS
C      IKEYWD(16.NI) THE SIZE OF THE SECONDARY KEY
```

```
      COMMON /TPTTFC/ IDITTF.DRITTF(243).LITTF
      INTEGER SKPLN,RWCHR,COLMX,FORMT(6,40)
      EQUIVALENCE (SKPLN.DRITTF(1)), (RWCHR.DRITTF(2))
      EQUIVALENCE (COLMX.DRITTF(3)), (FORMT(1,1).DRITTF(4))
C
C     IDITTF IS RECORD-ID OF DATA RECORD 1 OF TTF-TABLE.
C     DRITTF(243) IS USED TO HOLD CONTENTS OF DATA RECORD 1 OF TTF-TABLE.
C     LITTF IS LENGTH (IN CHARACTERS) OF DATA RECORD 1 OF TTF-TABLE.
C     SKPLN IS NUMBER OF LINES TO BE SKIPPED BETWEEN ROWS OF TABLE.
C     RWCHR IS CHARACTER TO BE USED IN PRINTING A SOLID ROW BETWEEN ROWS.
C     COLMX IS NUMBER OF COLUMNS (MAXIMUM COLUMN-ID) FOR TABLE.
C     FORMT(1.I) = PREFIX TYPE.  (A=1.I=2,D=3,B=4,C=5,SP=NEGATIVE).
C     FORMT(2.I) = PREFIX LENGTH.
C     FORMT(3.I) AND FORMT(4.I) = DATA TYPE AND LENGTH.
C     FORMT(5.I) AND FORMT(6.I) = SUFFIX TYPE AND LENGTH.
C     FORMT SPECIFIES THE COLUMN FORMATS.
C
C


C
C   JULY 1979
C
         COMMON  /USAGEC/ MONITR , NDSUSG
C
C   MONITR      0= NO COMMAND MONITORING
C               1=    COMMAND MONITORING
C   NDSUSG      DATA HANDLER DATASET NUMBER FOR  A1MONITR DATASET
C
C


C
C
C        JULY 22, 1977
      INTEGER R80CF,W80CF,W12CF,RTMCF,WLNCF
      COMMON /VFMT/
     1  R80CF(3),W80CF(4).ISU80C.LNCU.LDN1.LDN2,MCH(35).W12CF(6),ISU12C
     2, ISUTMC.RTMCF(3).ISULNC.WLNCF(3)
C           R80CF= FORMAT STATEMENT TO READ 80 CTRS INTO POSITIONS 1-80
C           W80CF= FORMAT.STATEMENT TO WRITE 80 CTRS ON ONE LINE
C           ISU80C= NUMBER OF SUS TO CONTAIN 80 CTRS
C           LNCU =LITERAL FOR NCU
C           LDN1.LDN2 = LITERAL FOR FORMAT TO PRINT 12 CTRS/3A4,/
C           MCH IS NUMBERS IN CHARACTER FORM.  MCH(1)=1H1: MCH(35)=2H35.
C           W12CF= FORMAT ST.TO WRT. 12CTRS STARTING AT LOCATION 1-120
C                  USER MUST PLACE THE NUMBER OF COLUMNS TO SKIP
C                  IN WORD 3 OF W12CF BY AN ASSIGNMENT FROM A VARIABLE
C                  DEFINED IN A DATA STATEEMNT, (OR USE THE ARRAY MC)
C           ISU12C= NUMBER OF SUS TO CONTAIN 12 CTRS
C           ISUTMC IS NUMBER OF WORDS TO STORE A TERMINAL LINE.
C           RTMCF IS FORMAT TO READ LINE FROM TERMINAL.
C           ISULNC IS 80/NCU (WHOLE WORDS NEEDED TO STORE AN OUTPUT LINE
C           FOR PRINTING ON A MAX. 80 CHAR DEVICE)
C           WLNCF IS FORMAT FOR ISULNC WORDS
C
C


C
C
C  JULY 24, 1979
C
         COMMON  /WIDOWC/ PFLAG
         INTEGER  PFLAG
C
C  PFLAG    SWITCH INDICATING
C                -1 =  SET BEFORE CALL TO WIDOW PROGRAM.INIT.IN BLOCK DATA
C                 0 =  SET BY WIDOW PROGRAM TO MEAN NORMAL PROCESSING
C                    LINE
C
C
```

3-8

DT